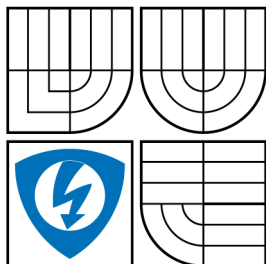


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## MĚŘICÍ VIDEOMIKROSKOP

MEASURING VIDEOMICROSCOPE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAKUB ČERNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ILONA KALOVÁ, Ph.D.



# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení:

Bytem:

Narozen/a (datum a místo):

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 53, Brno, 602 00

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

doc. Ing. Čestmír Ondrůšek, CSc., předseda oborové rady Silnoproudá

elektrotechnika a elektroenergetika

(dále jen „nabyvatel“)

### Čl. 1

#### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

☐ disertační práce

☐ diplomová práce

☒ bakalářská práce

☐ jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP:

Vedoucí/ školitel VŠKP:

Ústav:

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v\*:

☒ tištěné formě – počet exemplářů 1

☒ elektronické formě – počet exemplářů 1

---

\* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☐ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel

.....

Autor

## **Abstrakt**

Tato práce se věnuje propojení CCD kamery a mikroskopu a vytvoření tak univerzálního přístroje pro měření objektů pod mikroskopem.

V práci je podrobně řešen postup získání obrazu ze zdrojového hardwaru pomocí programu MATLAB, respektive jeho IMAGE ACQUISITION TOOLBOXu. Dále je řešeno předzpracování a segmentace obrazu.

V aplikaci jsou řešeny kalibrační postupy (na základě Houghovy transformace). Autonomní detekce a měření jednoduchých rovinných předmětů, složitějších za pomoci operátora.

## **Abstract**

This thesis focuses on the interconnection of CCD camera with microscope, which should lead to creation of universal tool for measuring objects under the scope.

In the following text there is detail description of the image acquisition process from source hardware using the MATLAB program, IMAGE ACQUISITION TOOLBOX respectively. Further there is also the description of preprocessing and segmentation of the image.

Calibration approaches are solved in the application (based on the Hough Transformation). The program is able to detect and measure basic flat objects, advanced with the help of operator.

## **Klíčová slova**

mikroskop; CCD kamera; zpracování obrazu; kalibrace; Houghova transformace

## **Keywords**

microscope; CCD camera; image processing; calibration, Hough Transformation

## **Bibliografická citace**

ČERNÝ, J. *Měřicí videomikroskop*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 60 s. Vedoucí bakalářské práce Ing. Ilona Kalová, Ph.D.

## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma Zařízení pro počítání kamenů jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ..... Podpis autora .....

## **Poděkování**

Děkuji vedoucí bakalářské práce Ing. Iloně Kalové, Ph.D. za kvalitní odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.



## **OBSAH**

<b>OBSAH.....</b>	<b>1</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>3</b>
<b>1. ÚVOD .....</b>	<b>5</b>
<b>2. PROSTŘEDKY PROGRAMU MATLAB.....</b>	<b>6</b>
2.1 Image Acquisition toolbox.....	6
2.1.1 Základní nastavení .....	7
2.1.2 Práce s daty .....	11
2.2 Image processing toolbox .....	11
2.2.1 Základní funkce .....	12
2.3 Grafické uživatelské prostředí .....	14
<b>3. SEGMENTACE.....</b>	<b>16</b>
3.1 Obraz.....	16
3.2 Segmentace prahováním .....	17
3.2.1 Prahování s globálním prahem .....	17
3.2.2 Prahování s množinou jasů .....	19
3.2.3 Poloprahování .....	19
3.3 Metody určování prahu .....	20
3.3.1 Histogram .....	20
3.4 Segmentace na základě detekce hran .....	21
3.4.1 Hranové detektory .....	21
3.4.2 Prahování obrazu hran .....	25
3.5 Houghova transformace.....	26
3.5.1 Detekce přímk .....	26
3.5.2 Detekce kružnic .....	28
<b>4. OSVĚTLENÍ.....</b>	<b>29</b>
4.1 Scéna bez osvětlení .....	29
4.2 Scéna osvětlená z boku žárovkou .....	30
4.3 Scéna osvětlená zespoda žárovkou .....	31
4.4 Nasvícení Kruhovým halogenovým světlem .....	32

<b>5. PROGRAM.....</b>	<b>33</b>
5.1 Nastavení programu .....	33
5.2 Získání obrazu.....	34
5.3 Úprava obrazu .....	35
5.4 Zpracování obrazu.....	37
5.4.1 Histogram .....	37
5.4.2 Řez X-Y .....	39
5.4.3 Uživatelské funkce.....	41
5.5 Měření .....	42
5.5.1 Kalibrace.....	42
5.5.2 Měření s uživatelem.....	45
5.5.3 Automatické měření.....	49
<b>6. ZÁVĚR.....</b>	<b>55</b>
<b>7. POUŽITÁ LITERATURA .....</b>	<b>56</b>
<b>8. PŘÍLOHY .....</b>	<b>57</b>

## SEZNAM OBRÁZKŮ

Obr. 2.1. Princip snímání obrazu .....	7
Obr. 2.2. GUI GUIDE builder.....	14
Obr. 3.1. Původní obrázek.....	17
Obr. 3.2. Vhodně naprahoovaný obrázek.....	18
Obr. 3.3. Vysoký práh .....	18
Obr. 3.4. Nízký práh.....	18
Obr. 3.5. Určování prahu z Histogramu .....	20
Obr. 3.6. Obrázek před použitím hranových detektorů.....	22
Obr. 3.7. Obrázek po aplikaci Robertsova operátoru.....	23
Obr. 3.8. Obrázek po aplikaci Prewitt operátoru .....	23
Obr. 3.9. Obrázek po aplikaci Sobelova operátoru .....	24
Obr. 3.10. Obrázek po aplikaci Cannyho operátoru.....	25
Obr. 3.11. Princip Houghovy transformace .....	26
Obr. 3.12. Vstupní obraz pro Houghovu transformaci.....	27
Obr. 3.13. Výstup Houghovy transformace .....	27
Obr. 4.1. Scéna bez přídavného osvětlení .....	29
Obr. 4.2. Scéna osvětlená žárovkovým světlem z boku.....	30
Obr. 4.3. Scéna osvětlená žárovkovým světlem zespoda.....	31
Obr. 4.4. Scéna osvětlená kruhovým halogenovým světlem na nejmenší výkon .....	32
Obr. 5.1. Programové prostředí.....	33
Obr. 5.2. Schéma použití programových funkcí .....	34
Obr. 5.3. Zdroj obrazu.....	35
Obr. 5.4. RGB obrázek.....	36
Obr. 5.5 GRAY obrázek.....	36
Obr. 5.6. BW obrázek vzniklý použitím Cannyho hranového detektoru.....	37
Obr. 5.7. Vývojový diagram funkce „Histogram“ .....	38
Obr. 5.8. Uživatelské prostředí pro funkci „Histogram“ pro RGB obraz.....	39
Obr. 5.9. Vývojový diagram funkce řezu X-Y.....	40
Obr. 5.10. Uživatelské prostředí funkce Posloupnosti.....	41

Obr. 5.11. Vhodný snímek kalibračního vzoru.....	43
Obr. 5.12. Nevhodné snímky kalibračního vzoru. ....	43
Obr. 5.13. Vývojový diagram funkce kalibrace.....	44
Obr. 5.14. Vývojový diagram funkce „Měření vzdáleností“ .....	45
Obr. 5.15. Uživatelské prostředí pro "Měření vzdáleností“ .....	46
Obr. 5.16. Měření úhlů.....	46
Obr. 5.17. Vývojový diagram měření úhlů .....	47
Obr. 5.18. Vývojový diagram měření kružnic .....	48
Obr. 5.19. Princip hledání kružnic .....	49
Obr. 5.20. Vývojový diagram hledání kružnic.....	50
Obr. 5.21. Původní obrázek pro detekci kružnic.....	51
Obr. 5.22. Nalezený střed a poloměr kružnice.....	51
Obr. 5.23. Princip hledání n-úhelníků.....	52
Obr. 5.24. Vývojový diagram hledání n-úhelníků .....	53
Obr. 5.25. Původní obraz pro detekci čtyřúhelníku .....	54
Obr. 5.26. Nalezený čtyřúhelník .....	54

## 1. ÚVOD

Tato práce se zabývá vytvořením uživatelského prostředí pro získání a další zpracování digitálního obrazu a navazuje na práci vytvořenou jako semestrální projekt.

Obraz získaný CCD kamerou umístěnou na mikroskopu SMZ 168-TL je nejprve zpracován televizní kartou v počítači a následně pomocí programu MATLAB a IMAGE ACQUISITION TOOLBOX přenesen do vlastního uživatelského prostředí, ve kterém jsou implementovány funkce pro další jeho zpracování.

Vlastní program je tedy vytvářen v GUI (prostředek pro vytváření uživatelského prostředí v programu MATLAB) a výtahem z IMAGE ACQUISITION a IMAGE PROCESSING TOOLBOXU.

## **2. PROSTŘEDKY PROGRAMU MATLAB**

Program MATLAB obsahuje řadu přídatných modulů (toolboxů) s předdefinovanými funkcemi pro snadnější práci programátora.

V této práci jsou použity dva toolboxy: IMAGE ACQUISITION a IMAGE PROCESSING TOOLBOX. Vlastní kapitolou je pak nástroj GUI pro vytváření uživatelského prostředí.

### **2.1 IMAGE ACQUISITION TOOLBOX**

IMAGE ACQUISITION TOOLBOX poskytuje funkce pro získávání obrázků a videí přímo do prostředí MATLAB a SIMULINK z PC-kompatibilních zařízení. Toolbox podporuje mnoho druhů hardwarů od nejlevnějších domácích kamer až po špičkové vědecké kamery.

Spolu s IMAGE PROCESSING TOOLBOXEM poskytuje kompletní vybavení pro práci s libovolnými obrazovými aplikacemi (získávání obrázků, videí, vizualizace dat, vyvíjení výrobních algoritmů, vytváření grafických uživatelských prostředí). Spolu s programem SIMULINK a VIDEO AND IMAGE PROCESSING BLOCKSET jsou využívány k simulaci a modelování obrazových aplikací v reálném čase.

Hlavní rysy toolboxu:

- Automatická detekce zařízení pro získávání obrazu
- Nastavování parametrů zařízení
- Živý náhled videa
- Získávání souvislého i statického obrazu
- Podpora mnoha výrobců zařízení
- Podpora Direct-X zařízení



Obr. 2.1. Princip snímání obrazu

### 2.1.1 Základní nastavení

V toolboxu je možné jednoduchými příkazy zjistit jaký hardware je k dispozici. V práci je uvedeno jen několik možných příkazů, které lze využít pro správu hardwaru.

Funkce „**imaqhwinfo('winvideo')**“ vrací strukturu, která obsahuje informace o všech dostupných video zařízeních.

**Syntaxe:**

**hwInfo =imaqhwinfo('winvideo')**

AdaptorDllName: [1x81 char]

AdaptorDllVersion: '1.10 (R2006a)'

AdaptorName: 'winvideo'

DeviceIDs: {[1]}

DeviceInfo: [1x1 struct]

Informace o konkrétním zařízení jsou obsaženy ve struktuře **hwInfo** a lze je zobrazit pomocí následující syntaxe: **Device = hwInfo.DeviceInfo(1).**

Struktura **Device** obsahuje informace:

- Implicitní rozlišení zařízení
- Podpora konfiguračních souborů
- Název zařízení
- ID zařízení (pro následné volání konstruktorů)
- Konstruktor přístupu k zařízení
- Pole podporovaných formátů obrazu

**Syntaxe:**

**Device = hwInfo.DeviceInfo(1)**

DefaultFormat: 'RGB32\_768x576'

DeviceFileSupported: 0

DeviceName: 'AVerMedia BDA Analog Capture'

DeviceID: 1

ObjectConstructor: 'videoinput('winvideo', 1)'

SupportedFormats: { 1x7 cell }

K přístupu ke kompletnímu seznamu aktuálního nastavení je možné využít funkce **get( )** s parametrem zdroje videa. Touto funkcí je možné například získat tyto informace:

**Příklad získaných hodnot**

- Jméno
- Formát obrazu
- Barevný typ obrazu
- Jména volajících funkcí

**Syntaxe:**

**vidobj = videoinput('winvideo', 1);**

**get(vidobj)**

Hodnoty aktuálně vybraného zdroje videa lze získat také pomocí funkce **getselectedsource( )**.



### Příklad získaných hodnot

- Typ zdroje videa (Tuner, S-video atd...)
- Analogový formát videa
- Světlost
- Kontrast
- Hloubka
- Nasycení
- Ostrost (pouze u podporovaného hardwaru)

### Syntaxe:

```
src = getselectedsource(vidobj);  
get(src);
```

### General Settings:

Parent = [1x1 videoinput]

Selected = on

SourceName = composite

Tag =

Type = videosource

### Device Specific Properties:

AnalogVideoFormat = ntsc\_m

Brightness = 128

Contrast = 68

HorizontalLocked = on

Hue = 0

Saturation = 64

Sharpness = 0

Funkce **get( )** dovoluje pouze získávat informace o daných proměnných. K jejich změně je potřeba využít funkce **set( )**. Pomocí této funkce není možné měnit všechny proměnné, pouze vybrané.

Nastavení hodnoty kontrastu (Contrast) na hodnotu 100 lze použít syntaxi:

**set(src, 'Contrast', 100);**

nebo alternativní „tečkový“ zápis:

**src.Contrast=100;**

U IMAGE ACQUISITION TOOLBOXU lze s výhodou využít tzv. „Callback“ funkcí a „Událostí“.

Události nastávají během získávání dat ve specifickém čase za splnění určitých podmínek.

#### **Typy Událostí**

- Error
- FramesAcquired
- Start
- Stop
- Timer
- Trigger

Callback funkce jsou takové funkce, které jsou vykonány v okamžiku, kdy nastane přidružená Událost.

#### **Typy Callback funkcí:**

- ErrorFcn
- FramesAcquiredFcn
- StartFcn
- StopFcn
- TimerFcn
- TriggerFcn

### 2.1.2 Práce s daty

Práce s daty je díky toolboxu velmi jednoduchá. Nejdříve je nutné nastavit zdroj videa. Funkcí **vidobj = videoinput('winvideo', 1, 'RGB\_768x576');** je vybrán zdroj z hardwaru s ID=1, rozlišením 768x576 a jsou použity ovladače z prostředí Windows.

Vybrané důležité funkce:

**preview(vidobj)** Vytvoří nové okno, ve kterém je zobrazeno aktuálně snímané video. Veškeré změny v nastavení se okamžitě projeví v promítaném videu. Vhodné pro nastavování světlosti, kontrastu atd.

**RGB=getsnapshot(vidobj)** Sejme jednotlivý snímek ze zařízení a uloží ho do proměnné RGB. Vhodné pro další zpracování obrazu.

**imagesc(RGB)** Zobrazí proměnnou RGB jako obrázek.

Pro ukládání získaného videa na disk je nutné změnit „LoggingMode“ z implicitního „memory“ na „disk“ nebo „disk&memory“ příkazem:

**set(vidobj, 'LoggingMode', 'disk')**

Aktuální stav proměnné lze zjistit příkazem:

**get(vidobj, 'LoggingMode')**

## 2.2 IMAGE PROCESSING TOOLBOX

IMAGE PROCESSING TOOLBOX poskytuje kompletní sadu normalizovaných algoritmů a grafických nástrojů pro zpracování obrazů, analýzu, vizualizaci a vývoj algoritmů. Lze obnovit zašlé nebo poničené obrázky, vylepšit srozumitelnost obrázků, zvýraznit rysy, analyzovat tvary, struktury.

Většina funkcí je napsána otevřeném jazyce MATLAB, což dává při programování možnost prozkoumat algoritmy, upravit zdrojový kód a vytvořit vlastní funkce.

Toolbox je využíván v oblastech jako biometrie, dálkové snímání, mikroskopy, testy polovodičů a další.

#### Hlavní rysy toolboxu:

- Vylepšování obrazu, včetně filtrování, zaostření a úprava kontrastu
- Analýza obrazu, detekce hran, morfologie, segmentace a měření
- Prostorové transformace
- Obrazové transformace včetně FFT, DCT a Radon
- Podpora víceprostorového zpracování
- Modulový interaktivní nástroje včetně ROI výběr, histogramy atd...
- Interaktivní obrazový a videový výstup
- DICOM vstup a výstup

#### 2.2.1 Základní funkce

IMAGE PROCESSING TOOLBOX obsahuje velké množství funkcí. V práci je uvedeno jen několik příkladů funkcí, které jsou využity pro vytvořený program.

**RGB=imread('jméno')** Načtení obrázku ze souboru s adresou 'jméno' a uložení do proměnné RGB. Je-li načítaný obrázek barevný, pak je proměnná RGB složena z trojrozměrného pole ['výška' 'šířka' 'rgb složka']. Hodnoty barev jsou od 0-255 a složka rgb má hodnoty 1 až 3 a ukazuje na jednotlivé spektrální složky barev:

- 1: červená
- 2: zelená
- 3: modrá

Vybrání konkrétní složky barvy lze pomocí syntaxe:

**Syntaxe:**

```
RGB=imread('obrazek.jpg');
rSlozka=RGB(:,:,1);
gSlozka=RGB(:,:,2);
bSlozka=RGB(:,:,3);
```

**Gray=rgb2gray(RGB);** Vytvoření šedotóného obrázku z obrázku barevného. Trojrozměrné pole dat se mění na pole dvojrozměrné. Hodnoty pixelů zůstávají na hodnotách 0-255.

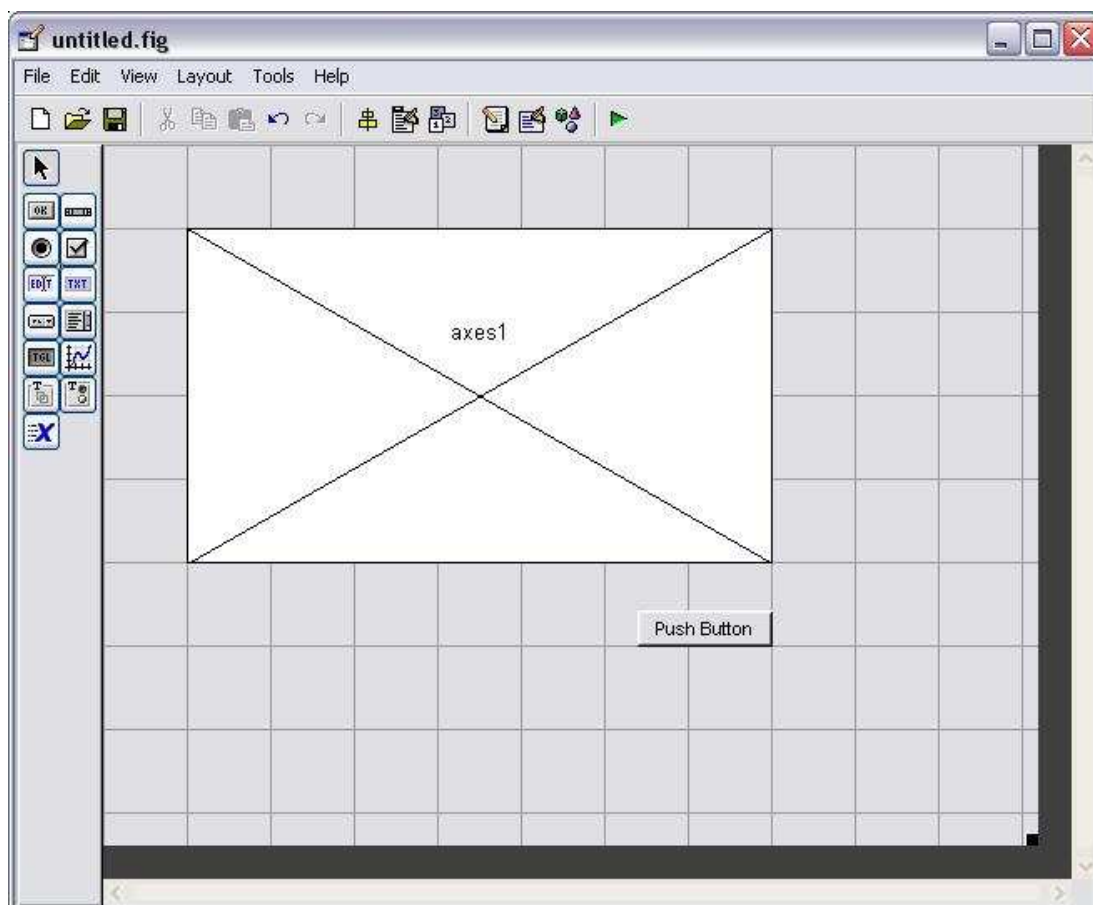
**Prah = graythresh(Gray);** Výpočet prahové hodnoty pro konverzi šedo-bílého obrázku a obrázek černobílý. Funkce používá k výpočtu Otsovu metodu. Hodnota proměnné „**Prah**“ leží v intervalu (0,1).

**CB = im2bw(Gray, Prah);** Převod šedého obrázku na obrázek černobílý. Jako prahová hodnota je využita proměnná „Prah“. Výsledný obrázek obsahuje hodnoty 0 pro černou a 1 pro barvu bílou.

**CB = bwareaopen(CB, N);** Odstranění všech objektů menších než N pixelů. Vhodné pro odstraňování šumu apod.

**imshow(CB);** Zobrazení obrázku.

## 2.3 GRAFICKÉ UŽIVATELSKÉ PROSTŘEDÍ



*Obr. 2.2. GUI GUIDE builder*

Grafické uživatelské prostředí (GUI) je zobrazovač, který obsahuje zařízení nebo komponenty, které umožní uživateli vykonávat interaktivní úkoly. Pro vykonání těchto úkolů nemusí uživatel vytvářet skripty (M-file) nebo psát příkazy do příkazové řádky MATLABU. Taktéž nemusí být uživatel seznámen s detaily různých funkcí, ale jen je využívá.

Součástí GUI mohou být různé komponenty (různá menu, nástrojové lišty, posuvné seznamy a posuvníky). Data lze zobrazovat ve formě tabulky nebo grafu.

Každá komponenta, a GUI sám, jsou přidruženy s jednou nebo více uživatelsky psaných rutin zvané jako „Callback“. Spuštění každého Callbacku spočívá ve vykonání jednotlivých uživatelských akcí, jako jsou zmáčknutí tlačítka, kliknutí myši, výběr položky v menu nebo umístěním kurzoru na komponentu. Programátor GUI pak již programuje události, které následují po zavolání určitého callbacku.

GUI lze programovat dvěma způsoby.

- Pomocí GUIDE builder
- Programovatelně - veškeré komponenty jsou poté vytvořeny pomocí speciálních funkcí.

Při programování lze využít předdefinované funkce pro vytvoření standardních dialogových oken, jako je okno pro uložení či načtení souboru.

### 3. SEGMENTACE

Stěžejní část zpracování obrazu je segmentace obrazu. Segmentací obrazu se rozumí analýza obrazu, která vede k nalezení a rozdělení objektů ve zpracovávaném obraze na části, které mají spojitost se skutečnými objekty. Výsledkem této operace je rozdělení obrazu na jednotlivé nepřekrývající se části, které buď jednoznačně korespondují s objekty ve vstupním obraze, pak se jedná o **kompletní segmentaci**, nebo vytvořené části nemusí přímo odpovídat objektům ve vstupním obraze a pak se jedná o **částečnou segmentaci**.

Při částečné segmentaci je výsledkem rozdělení obrazu do samostatných částí, které jsou homogenní vzhledem k určitým zvoleným vlastnostem, jako jsou jas, barva apod. Po zpracování složitého obrazu je výsledkem seznam oblastí, které jsou homogenní v jistých zvolených rysech. Tyto oblasti se mohou překrývat.

[4]

#### 3.1 OBRAZ

Obraz je reprezentován jako spojité rozložení jasu (intenzity) v rovině. Rozložení je definováno obrazovou funkcí:

$f(x, y)$  statický obraz

$f(x, y, t)$  dynamický obraz

kde **f** je reálná spojitá funkce s reálnými argumenty (**x,y** jsou souřadnice bodů v rovině a **t** hodnota času).

Při zpracování je obraz vždy diskretizován a lze jej tak považovat za nespojitou funkci  $f(x,y)$ .



## 3.2 SEGEMENTACE PRAHOVÁNÍM

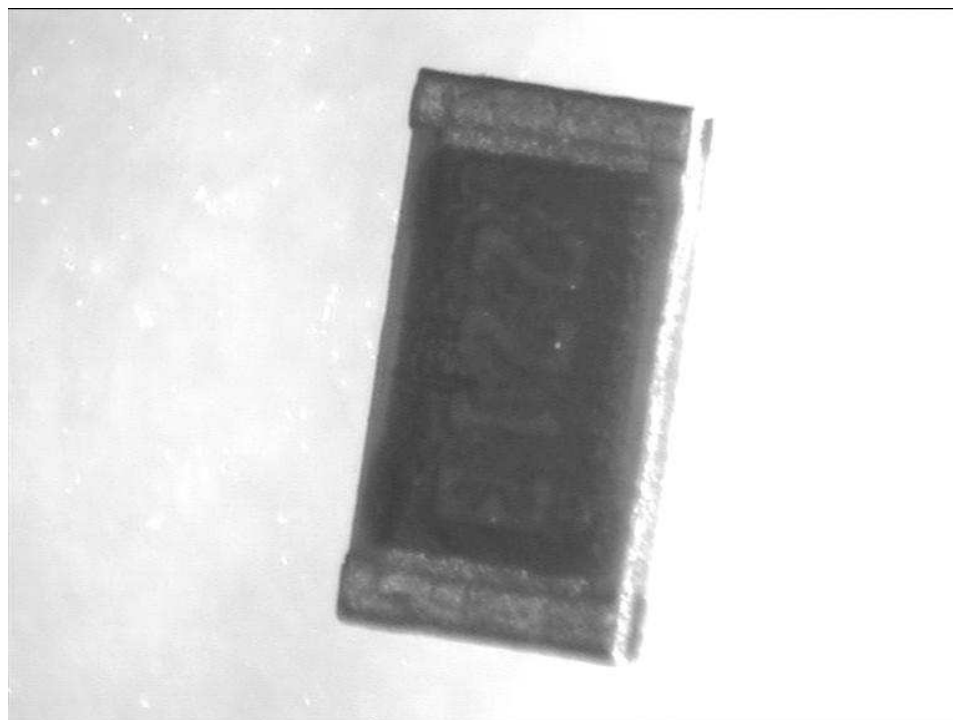
Tato metoda vychází ze skutečnosti, že mnoho objektů a oblastí obrazu je charakterizováno konstantními vlastnostmi povrchu (odrazivost, absorpční vlastnosti). Platí-li tento předpoklad, pak je možné použít určitou jasovou konstantu  $T$  jako prahu k oddělení objektů od pozadí. Tato metoda je velice jednoduchá a lze ji využívat i v aplikacích pracujících v reálném čase.

### 3.2.1 Prahování s globálním prahem

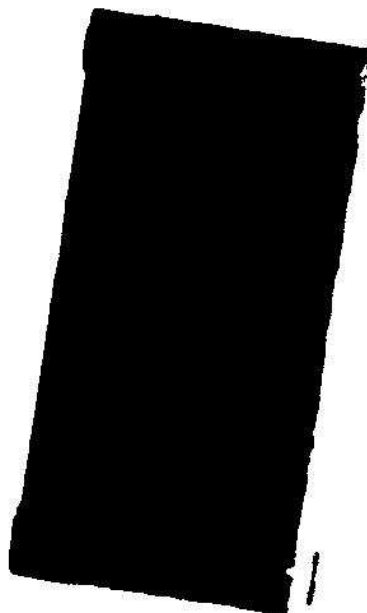
Prahování je transformace vstupního obrazu  $f(x,y)$  na výstupní binární obraz  $g(x,y)$  podle vztahu:

$$g(x, y) = \begin{cases} 1, & \text{...} f(x, y) \geq T \\ 0, & \text{...} f(x, y) < T \end{cases}$$

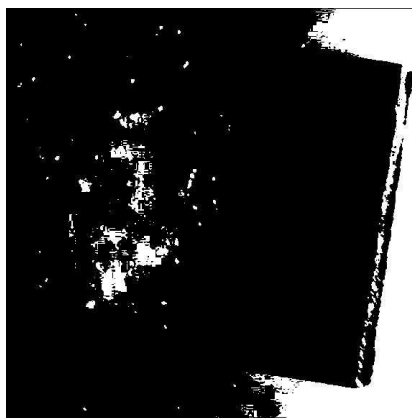
kde  $T$  je určená prahová konstanta a  $f(x,y)$  je vstupní obrazová funkce.



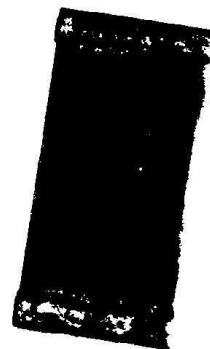
*Obr. 3.1. Původní obrázek*



*Obr. 3.2. Vhodně naprahovaný obrázek*



*Obr. 3.3. Vysoký práh*



*Obr. 3.4. Nízký práh*

Pro správný výsledek je důležitá správná volba prahové hodnoty **T**. Volbu této konstanty lze určovat interaktivně nebo použitím některé metody pro automatické určení prahové hodnoty.

Metoda prahování s globálním prahem nemusí ani v relativně jednoduchých scénách dávat požadované segmentační výsledky. To je zapříčiněno změnami jasů objektů i pozadí. Tyto změny mohou být zapříčiněny např. nerovnoměrností osvětlení, nestejnými vlastnostmi snímače apod. [4]

Možným řešením tak je **prahování s proměnným prahem**.

### 3.2.2 Prahování s množinou jasů

Tato metoda je založena metodě prahování s globálním prahem. Místo jediné konstanty  $T$  je určena množina jasů  $D$ . Jasy náležící množině  $D$  jsou považovány za objekty a ostatní je považováno za pozadí

$$g(x, y) = \begin{cases} 1, \dots, f(x, y) \in D \\ 0, \dots, \text{ostatní} \end{cases}$$

Tato metoda lze využít pro hledání hranic objektů.

### 3.2.3 Poloprahování

Při vizuálním hodnocení výsledků člověkem lze využít takovou definovanou podmnožinu  $D$  pro kterou platí:

$$g(x, y) = \begin{cases} f(x, y), \dots, f(x, y) \in D \\ 0, \dots, \text{ostatní} \end{cases}$$

Takto lze získat obraz, kde bylo odstraněné pozadí při zachování informací o jasu v objektech. Výsledkem je obraz s úplnou jasovou informací.

### 3.3 METODY URČOVÁNÍ PRAHU

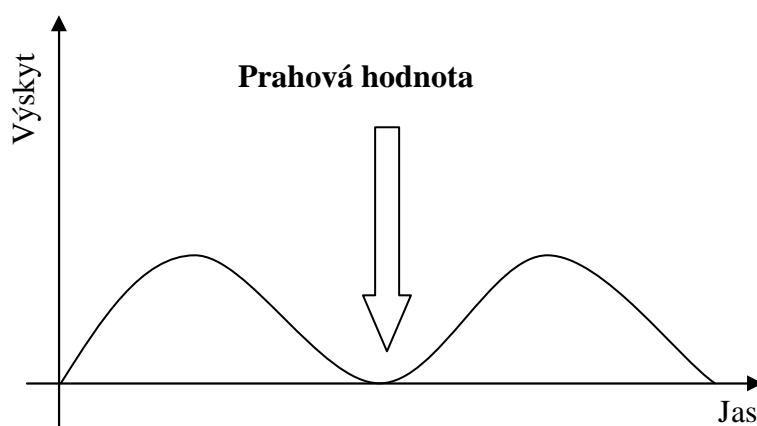
Jsou-li známy vlastnosti, které má mít výsledný obraz, určíme práh tak, aby byla tato vlastnost segmentováním splněna.

S využitím apriorní znalosti poměru ploch objektů a pozadí lze na základě histogramu určit takovou hodnotu prahu  $T$ , aby byl splněn zadaný poměr. Tato metoda se nazývá **procentní prahování**. V praxi však apriorní znalost procentuálního zastoupení není k dispozici. [4]

#### 3.3.1 Histogram

Jsou-li v obraze objekty podobného jasu a odlišné od pozadí je histogram dvouvrcholový (bimodální). Jeden vrchol odpovídá četnosti obrazových elementů objektů a druhý odpovídá pozadí.

Vhodná prahová hodnota leží v minimu mezi oběma vrcholy a měla by splňovat požadavek na minimální segmentační chyby. V případě že histogram je vícevrcholový (multimodální), lze určit prahovou hodnotu v minimech mezi dvěma maximy.



Obr. 3.5. Určování prahu z Histogramu

### 3.4 SEGMENTACE NA ZÁKLADĚ DETEKCE HRAN

Další možností segmentace obrazu jsou metody založené na informacích o hranách obrazu. Tento přístup byl v omezené formě jeden z prvních postupů v segmentaci obrazu.

Segmentace vychází ze skutečnosti, že objekty v obraze jsou ohraničeny hranami, které jsou nalezeny některým z hranových operátorů. [4]

#### 3.4.1 Hranové detektory

Hrana v obraze je vektorová veličina a je určena velikostí a směrem. Tyto veličiny vycházejí z gradientu  $\nabla$  obrazové funkce v obrazovém elementu. Hrana indikuje body obrazu, ve kterých dochází ke změnám obrazové funkce.

Hranové operátory se dělí do dvou skupin:

- Metody založené na první derivaci
- Metody založené na druhé derivaci

##### 3.4.1.1 Metody založené na první derivaci

Tyto metody aproximují derivace obrazové funkce pomocí diferencí realizovaných diskretní konvolucí. Směr gradientu  $\nabla$  se odhaduje hledáním masky  $\mathbf{h}$ , která odpovídá největší velikosti gradientu.

**Roberts operátor:**

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

### Prewitt operátor:

Prewitt operátory 3x3 a 5x5:

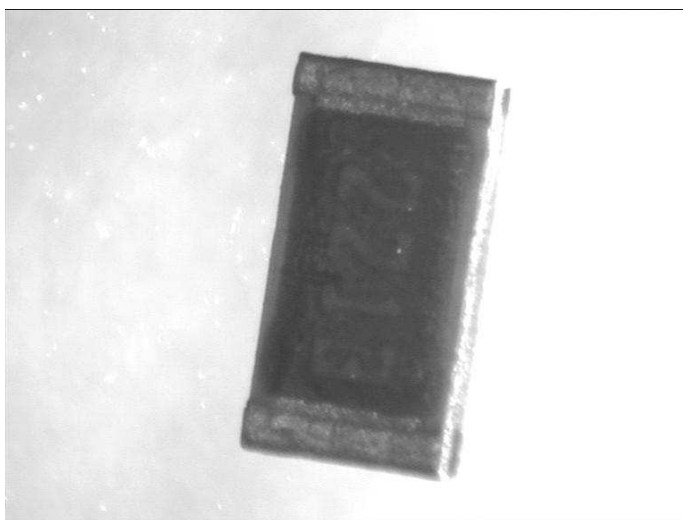
$$h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$h_y = \begin{bmatrix} -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

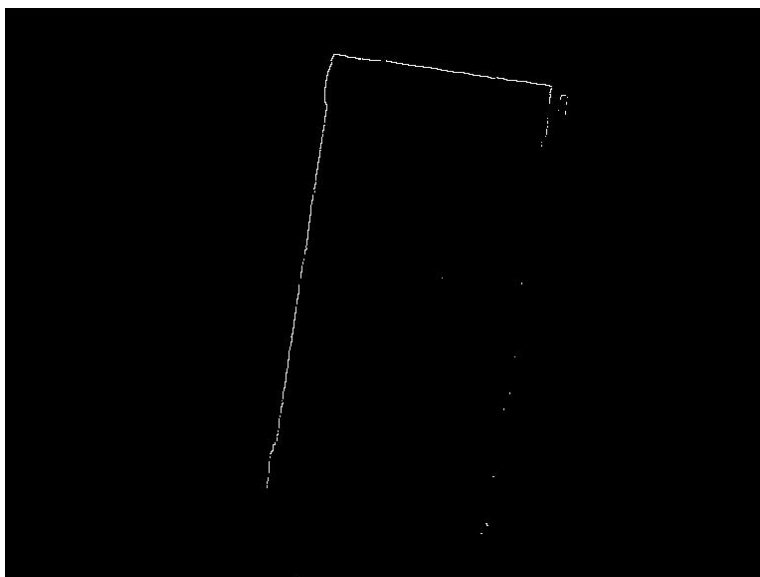
### Sobel operátor:

$$h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

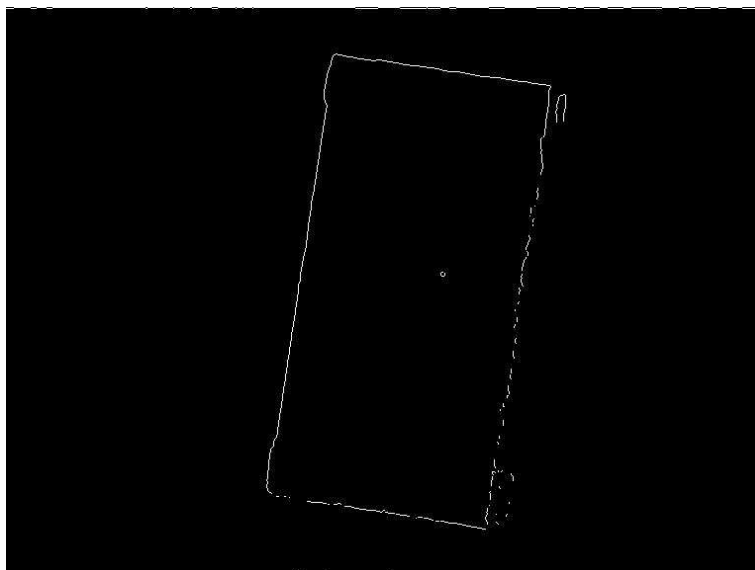
$$h_x = (h_y)^T$$



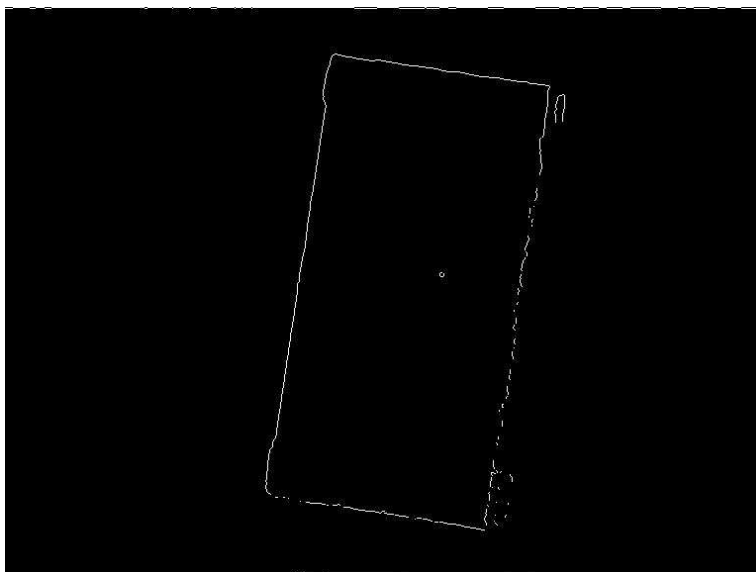
Obr. 3.6. Obrázek před použitím hranových detektorů



*Obr. 3.7. Obrázek po aplikaci Robertsova operátoru s automatickou volbou parametrů*



*Obr. 3.8. Obrázek po aplikaci Prewitt operátoru s automatickou volbou parametrů*



*Obr. 3.9. Obrázek po aplikaci Sobelova operátoru s automatickou volbou parametrů*

#### 3.4.1.2 Metody založené na druhé derivaci

Tato skupina metod spočívá v hledání hran obrazu v takových místech, kde druhá derivace obrazové funkce prochází nulou.

##### **Canny metoda:**

Základním principem této metody je hledání průchodu druhé derivace obrazové funkce nulou.

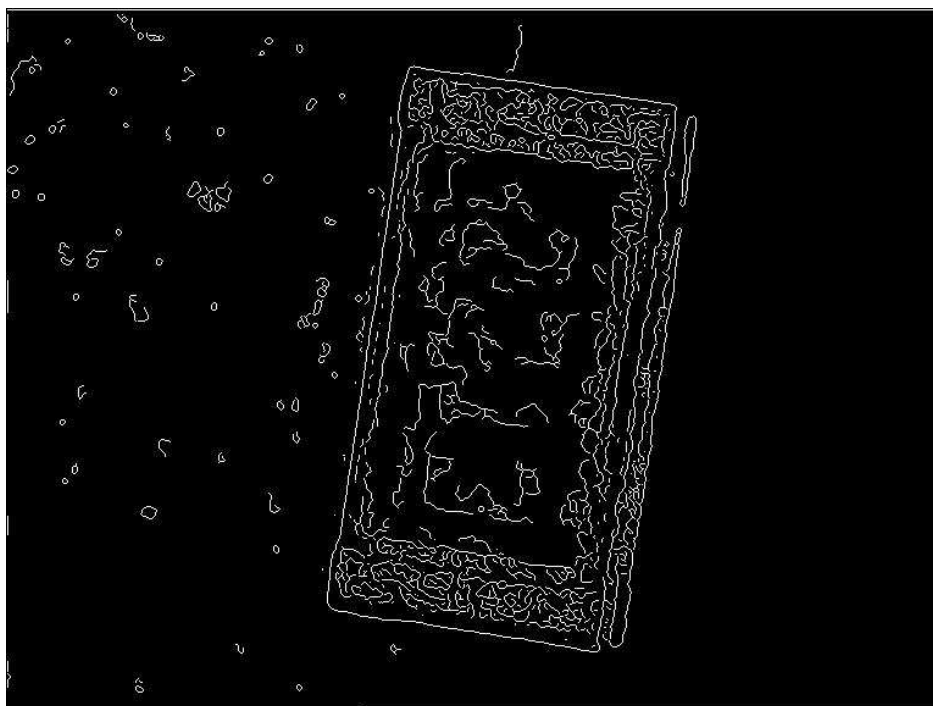
Požadavky při detekci hran

- Detekční kritérium ( neopomíjení významných hran)
- Lokalizační kritérium ( minimální rozdíl mezi skutečnými a detekovanými hrany)
- Jednoznačnost hran ( žádná hrana nesmí být detekována dvakrát)



Potup při aplikaci Cannyho hranového detektoru:

- Eliminace šumu (použití filtrů)
- Výpočet gradientu  $\nabla$
- Nalezení lokálních extrémů
- Eliminace nevýznamných hran



*Obr. 3.10. Obrázek po aplikaci Cannyho operátoru s automatickou volbou parametrů*

### 3.4.2 Prahování obrazu hran

Malé velikosti hran nemusejí odpovídat jen skutečným hranám, ale i šumu vzniklém při digitalizaci obrazu. K odstranění těchto nevýznamných hran je použito prahování hran s vhodnou prahovací konstantou **T**. Je-li znám poměr výskytu objektů k pozadí je možné využít procentní prahování. Využít lze i veškeré znalosti vlastnostech snímaného objektu (délka, šířka apod.).

[4]

### 3.5 HOUGHOVA TRANSFORMACE

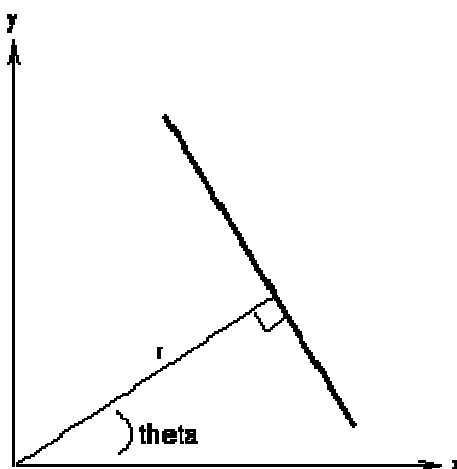
Houghova transformace je metoda, která vyhledává analyticky popsané objekty. Pro velkou náročnost na výpočetní výkon PC je tato metoda vhodná pro vyhledávání relativně jednoduchých objektů jako jsou přímky, kružnice a elipsy.

Pro všechny vstupní hodnoty (v binárním obraze body s jasovou hodnotou 1) jsou hledány všechna možná řešení hledaných křivek. S každým bodem je inkrementován Houghův prostor. Parametry hledaného objektu pak odpovídají indexům buňky Houghova prostoru s nejvyšší naakumulovanou hodnotou.

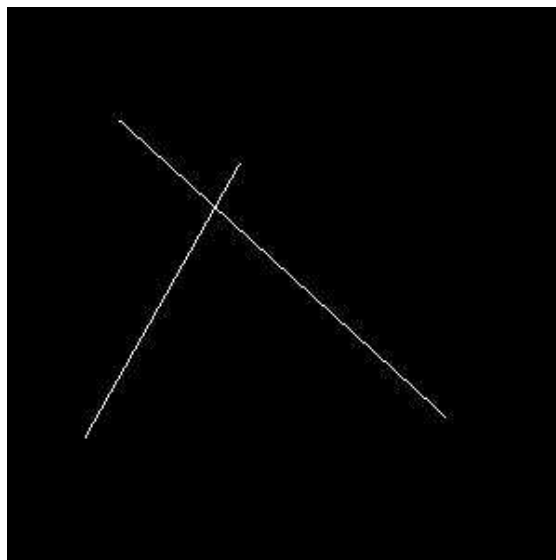
#### 3.5.1 Detekce přímek

Přímka v rovině je popsána rovnicí  $r = x \cdot \cos \theta + y \cdot \sin \theta$ , kde  $r$  je velikost normály přímky od počátku a  $\theta$  je úhel, který svírá normála a osa  $x$ . Pro každý vstupní bod  $(x, y)$  jsou nalezena všechna možná řešení  $(r, \theta)$  a inkrementován prostor  $H(r, \theta)$ .

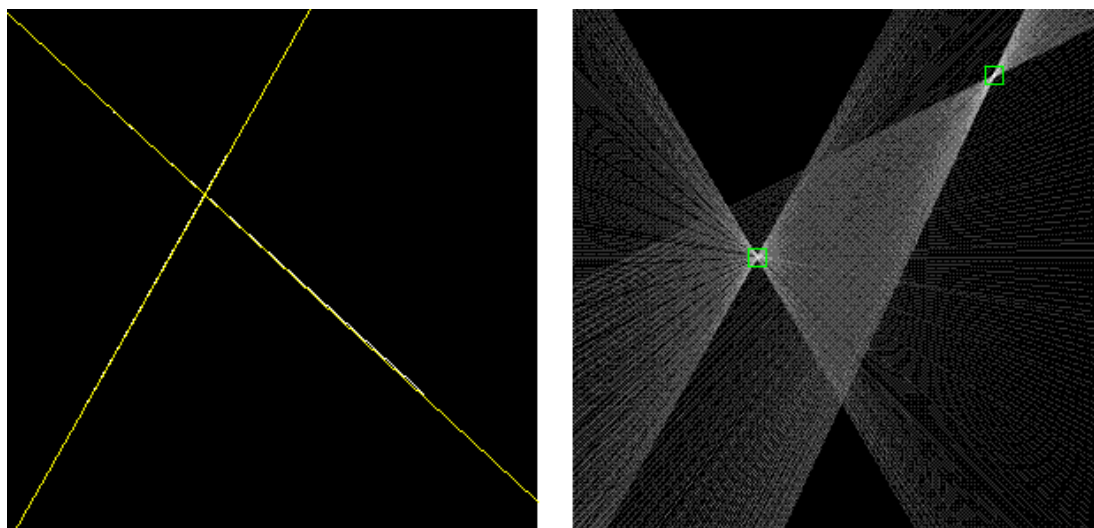
[2]



Obr. 3.11. Princip Houghovy transformace



*Obr. 3.12. Vstupní obraz pro Houghovu transformaci*



*Obr. 3.13. Výstup Houghovy transformace (vlevo detekované přímky, vpravo Houghův prostor)*

### 3.5.2 Detekce kružnic

Kružnice v rovině je popsána rovnicí  $r = (x - a)^2 + (y - b)^2$ , kde  $r$  je poloměr hledané kružnice a  $a, b$  jsou souřadnice středu kružnice. Pro každý vstupní bod  $(x, y)$  jsou nalezeny všechny možné řešení  $(a, b, r)$  a inkrementován prostor  $H(a, b, r)$ . Je-li hledána kružnice se známým poloměrem, je Houghův prostor dvojrozměrný, s neznámým poloměrem je Houghův prostor rozšířen do třetí dimenze.

Pro hledání středu kružnice  $(a, b)$  o daném poloměru  $r$  je vhodné použít vzorce s postupnou inkrementací úhlu  $\varphi$ .

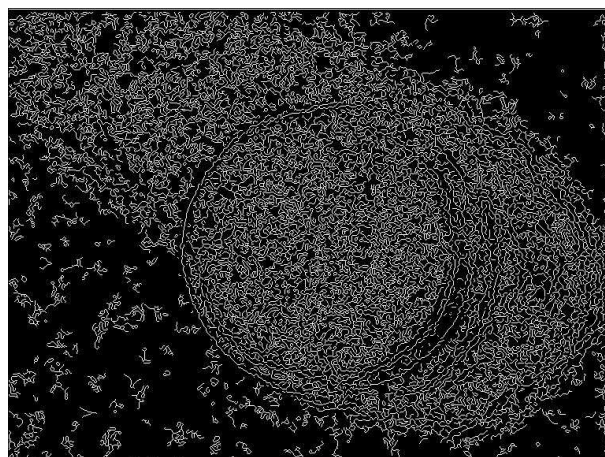
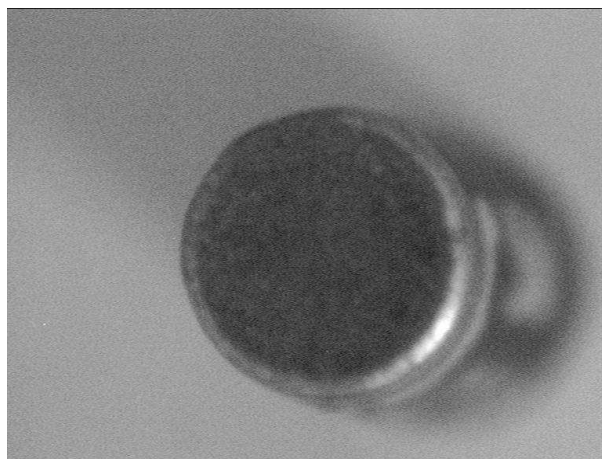
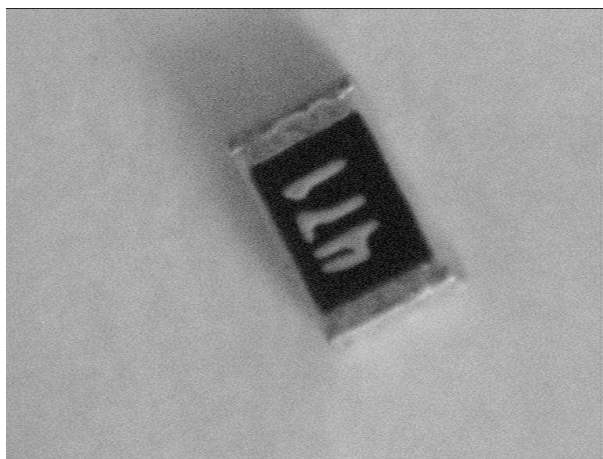
- $a = x - r \cos(\varphi / 180 \pi)$
- $b = y + r \sin(\varphi / 180 \pi)$

## 4. OSVĚTLENÍ

Při získávání snímku zaujímá nasvícení důležitou roli. Rozhoduje jak typ osvětlení tak i jeho intenzita a směr. Při nevhodném osvětlení mohou zaniknout potřebné informace o obrazu a nelze je žádnými úpravami vrátit zpět.

### 4.1 SCÉNA BEZ OSVĚTLENÍ

Scéna bez použití přídavného světla je velice zašuměná, hrany jsou neostře a hranové detektory mají problém tyto hrany nalézt. Pro další zpracování je tato scéna nevhodná.

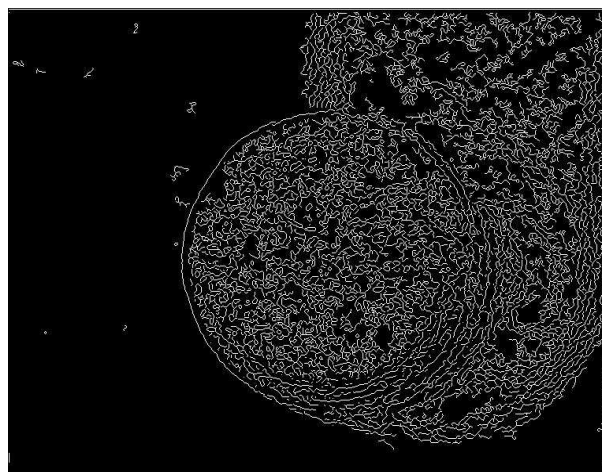
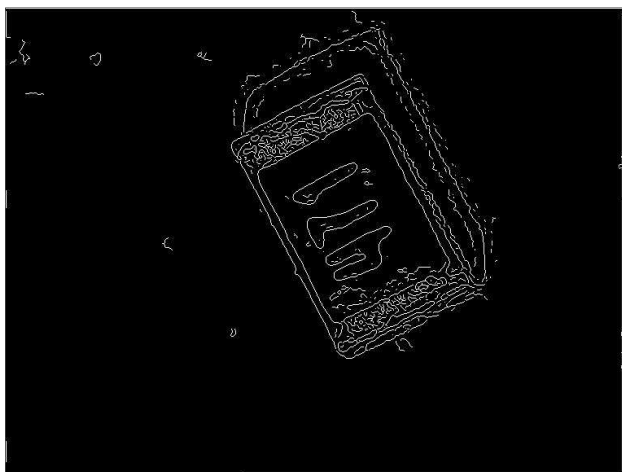
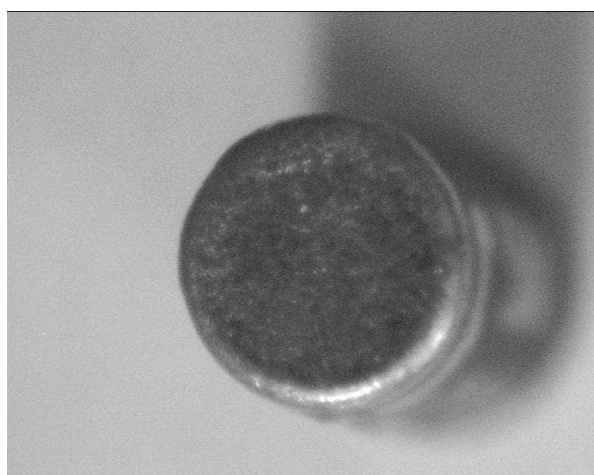
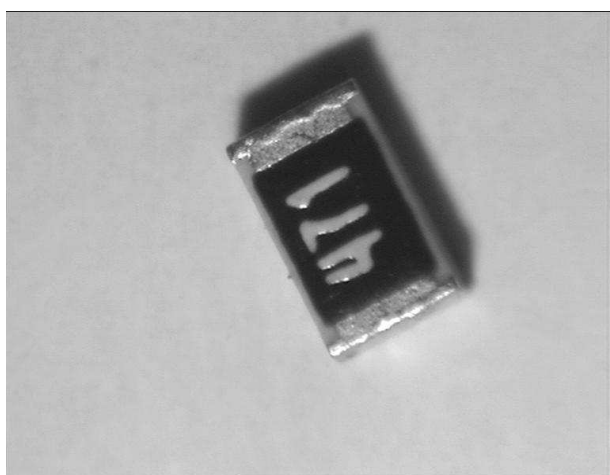


*Obr. 4.1. Scéna bez přídavného osvětlení (nahore) obraz po aplikaci Cannyho  
hranového detektoru (dole)*



## 4.2 SCÉNA OSVĚTLENÁ Z BOKU ŽÁROVKOU

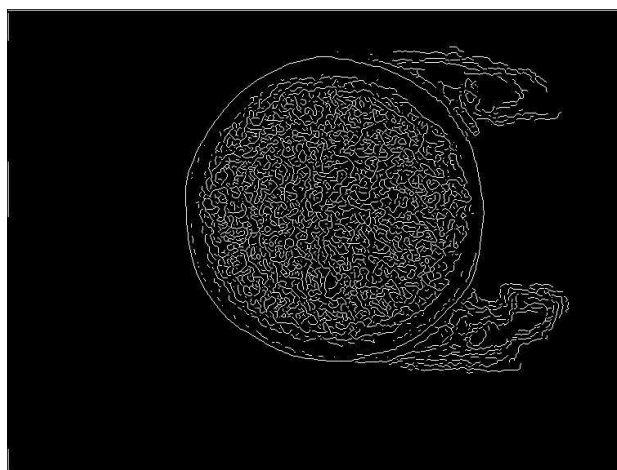
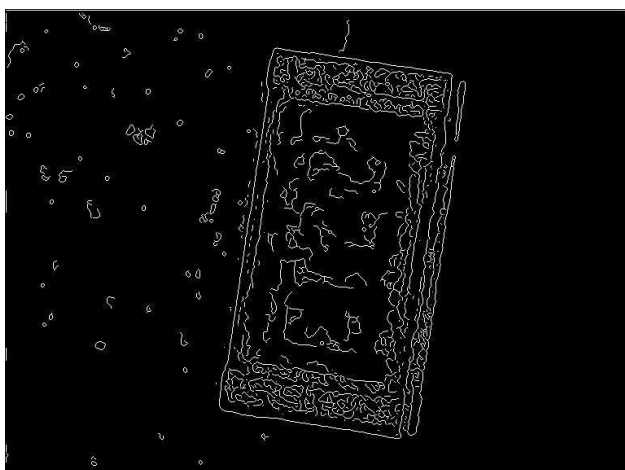
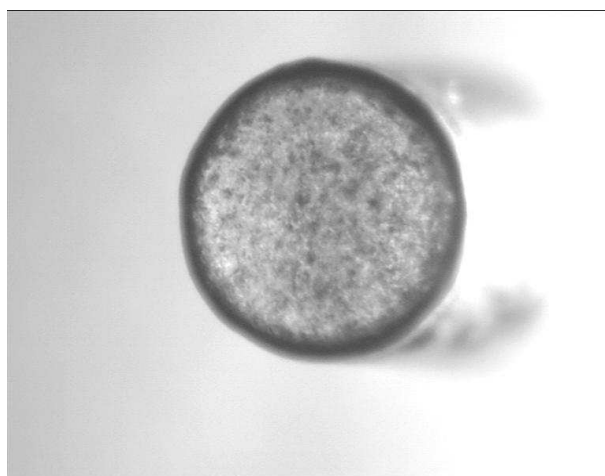
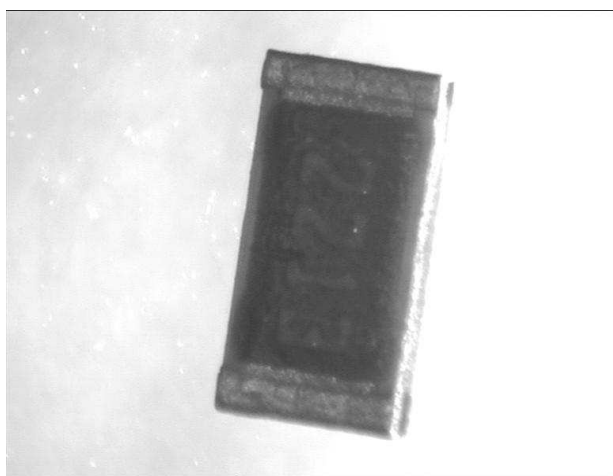
Scéna je bez velkého zašumění. Snímaný objekt má ostré hrany, ale vrhá hutný stín, který je detekován jako hrana. Takto nasvícená scéna je vhodná jen pro rovinné obrazy.



*Obr. 4.2. Scéna osvícená žárovkovým světlem z boku (nahore) obraz po aplikaci  
Cannyho hranového detektoru (dole)*

### 4.3 SCÉNA OSVĚTLENÁ ZESPODA ŽÁROVKOU

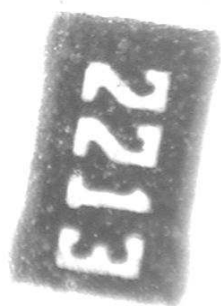
Scéna je bez velkého zašumění. Snímaný objekt má ostré hrany a nevrhá stín. Tento způsob nasvícení se jeví jako nejlepší možné řešení za daných podmínek.



*Obr. 4.3. Scéna osvětlená žárovkovým světlem zespoda (nahore) obraz po aplikaci Cannyho hranového detektoru (dole)*

#### 4.4 NASVÍCENÍ KRUHOVÝM HALOGENOVÝM SVĚTLEM

Kvůli nevyhovující regulaci intenzity světla je tento způsob osvětlení scény zcela nepoužitelný. Na nejmenší výkon světla zanikají hrany objektů a na maximální výkon zanikají objekty celé.



*Obr. 4.4. Scéna osvětlená kruhovým halogenovým světlem na nejmenší výkon*



## 5. PROGRAM

Program je primárně určen k použití do laboratoře k práci s mikroskopem SMZ 168-tl a CCD kamerou. Avšak tento program lze využít pro jakékoliv aplikace, kde je nutné získávat a zpracovávat obraz. K tomu také byly přizpůsobeny všechny jeho funkce a především pro větší flexibilitu byly přidány Uživatelské funkce.



Obr. 5.1. Programové prostředí

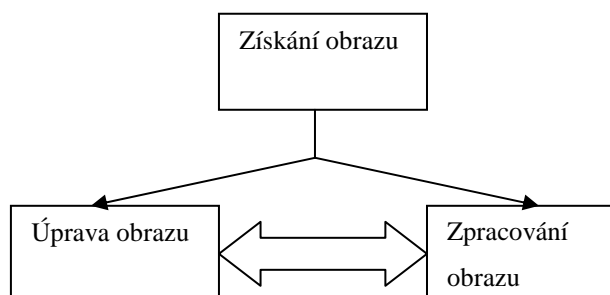
### 5.1 NASTAVENÍ PROGRAMU

Programové funkce jsou rozděleny do tří hlavních skupin.

- Získání obrazu
- Úprava obrazu

- Zpracování obrazu

Nejdříve musí uživatel použít funkce pro získání obrazu (snímek kamery, načtení obrázku) a teprve poté mu program umožní použít libovolnou kombinaci úpravy a zpracování obrázku.



*Obr. 5.2. Schéma použití programových funkcí*

Program umožňuje uživateli pracovat s třemi datovými druhy obrazu: RGB pro barevné obrázky, GRAY pro šedé a BW pro binární obraz.

## 5.2 ZÍSKÁNÍ OBRAZU

Program je nastaven tak, aby podporoval veškerý hardware, pro který jsou k dispozici ovladače systému Windows. V případě přítomnosti více zařízení, které jsou schopny zpracovat obraz, může si uživatel vybrat ze seznamu dostupných zařízení. Možnosti nastavení zdroje obrazu:

- Zdrojový hardware
- Rozlišení
- Zdroj signálu
- Barevný typ obrazu.

Po nastavení zdroje obrazu je možné použít náhled, ve kterém uživatel zjišťuje aktuální nastavení zdrojového signálu.

Pomocí posuvníku a editačního okna lze nastavovat tyto parametry:

- Jas
- Kontrast
- Hloubka
- Saturace



*Obr. 5.3. Zdroj obrazu*

Některé jiné karty na zpracování umožňují měnit i jiné parametry. Na změnu těchto parametrů je pak zapotřebí použít uživatelských funkcí.

Samotné získání obrazu je provedeno po stisknutí tlačítka „Snímek“. Podle zdroje je obrázek uložen do náležitých proměnných (RGB, GRAY).

Obrázek je také možno načíst ze souboru uloženého v počítači.

### 5.3 ÚPRAVA OBRAZU

Úprava obrazu spočívá především v převodu mezi formáty obrazu a úpravou binárního obrazu.

Pro RGB obraz jsou implementovány tyto funkce:

- Převod do GRAY

- Převod do BW pomocí prahování (hodnota 0 pro automatické zvolení prahu)

Pro GRAY obraz jsou implementovány tyto funkce:

- Převod do BW pomocí prahování (hodnota 0 pro automatické zvolení prahu)
- Převod do BW pomocí hranových detektorů

Jako defaultní hranový detektor je použit „Canny“ metoda. Uživatel si může zvolit i jiný z těchto možností:

- Sobel
- Prewitt
- Roberts
- Laplacian of Gaussian

Pro BW obraz jsou implementovány tyto funkce:

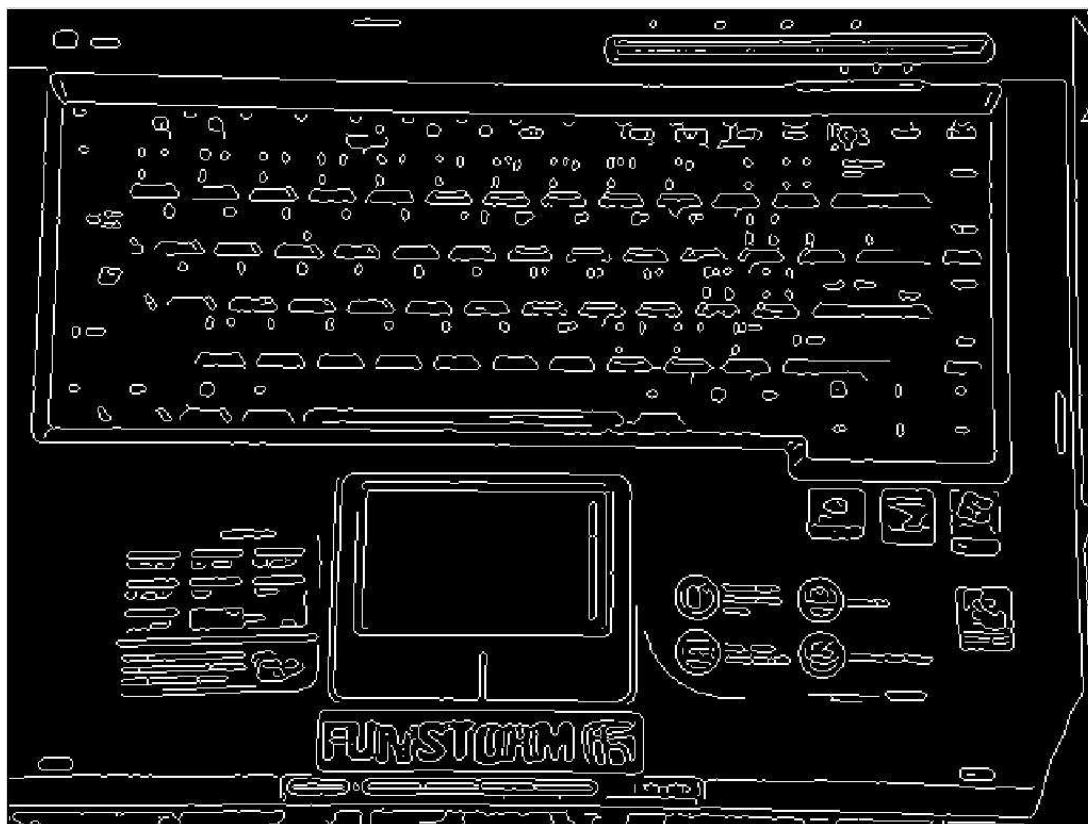
- Odstranění objektů menších než uživatelem zadané číslo
- Odstranění uživatelem zvolené plochy
- Invertování obrazu



Obr. 5.4. RGB obrázek



Obr. 5.5 GRAY obrázek



*Obr. 5.6. BW obrázek vzniklý použitím Cannyho hranového detektoru*

## 5.4 ZPRACOVÁNÍ OBRAZU

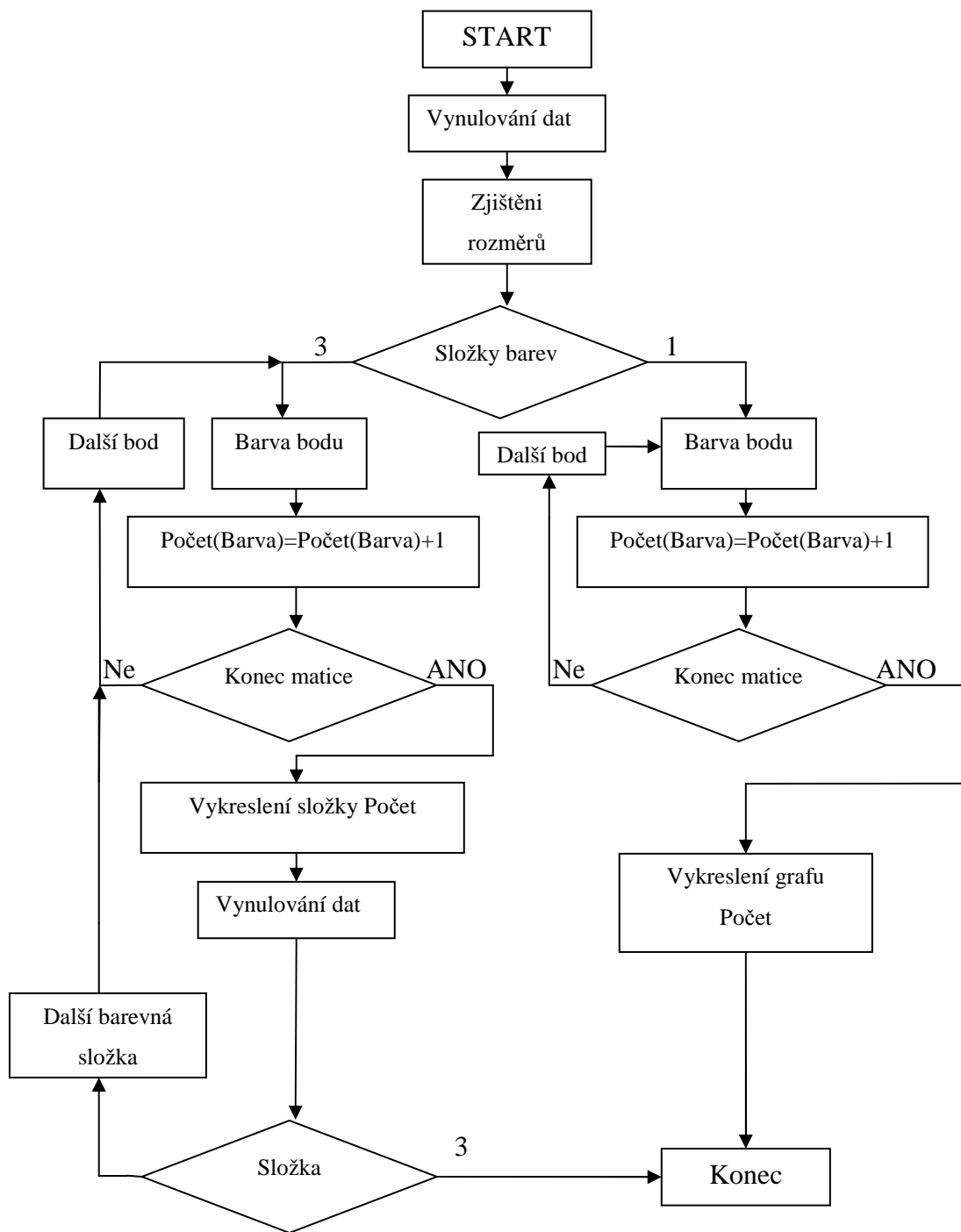
V programu jsou implementovány některé důležité funkce, které buď pomáhají uživateli zpracovávat obraz nebo jej zpracovávají automaticky.

### 5.4.1 Histogram

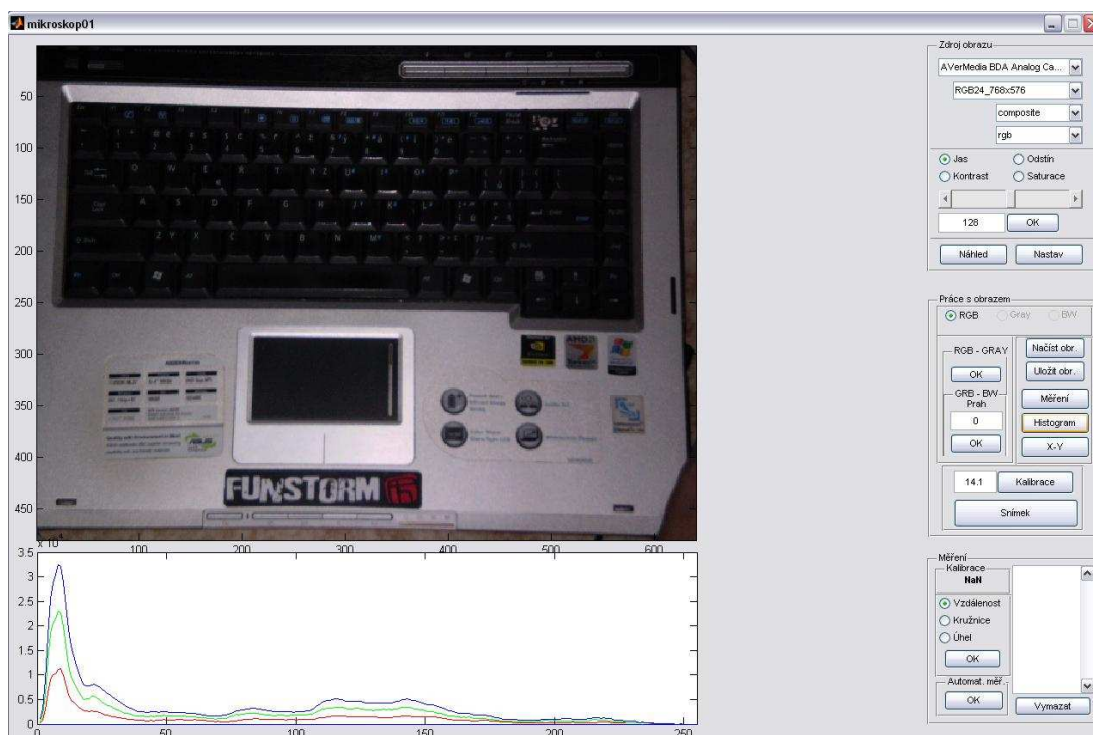
Histogram je funkce, která vypočítá počet výskytů pixelu se stejnou barvou. Tato funkce je vhodným prostředkem pro určování hodnoty prahu při segmentaci.

Je-li obraz složen ze třech složek barev vypočítá program výskyt pro všechny složky barev. Výsledek je zobrazen v grafu, kde barva složky odpovídá barvě grafu.

Funkce je umístěna v souboru „fce\_cetnost.m“.



Obr. 5.7. Vývojový diagram funkce „Histogramu“

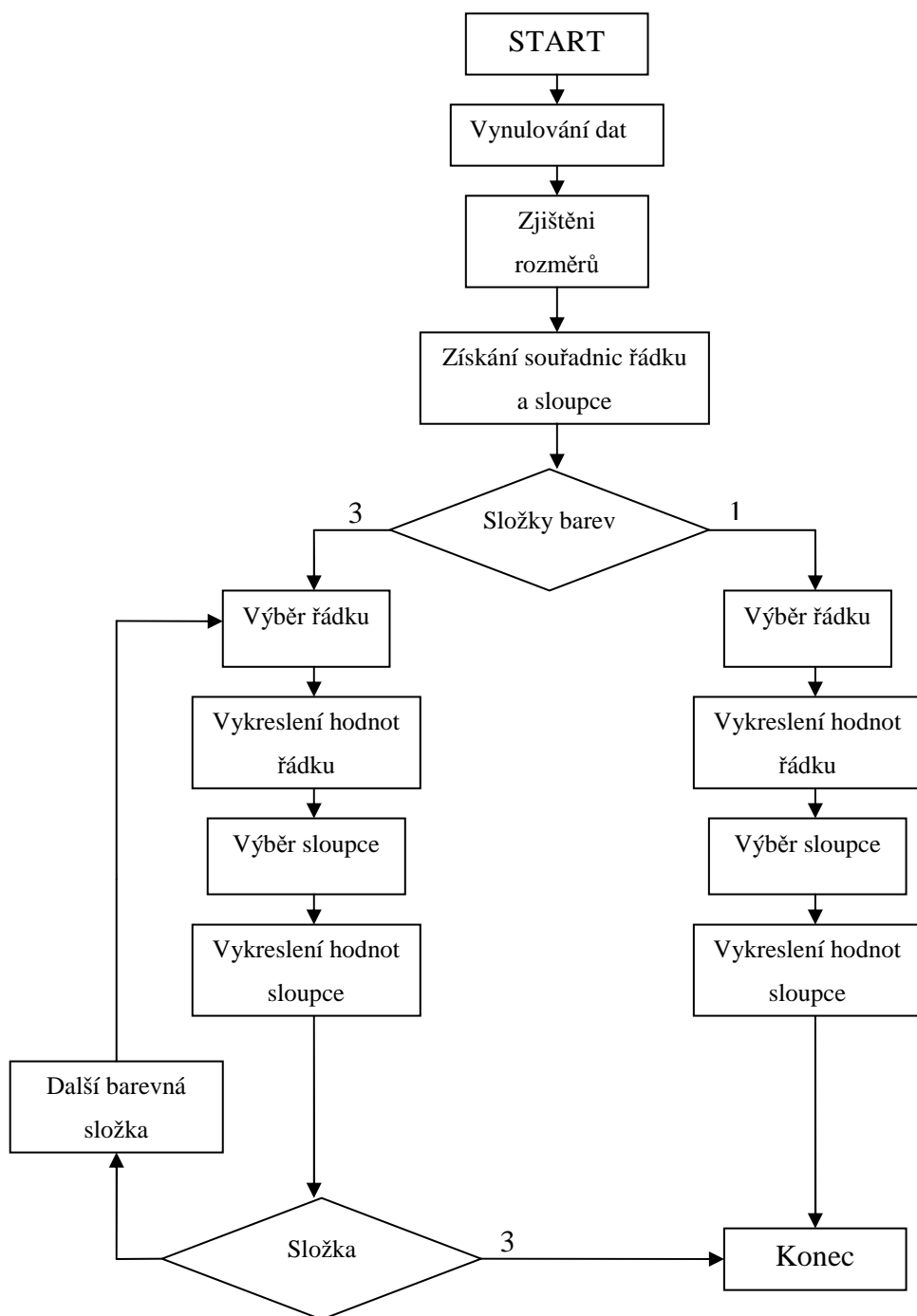


Obr. 5.8. Uživatelské prostředí pro funkci „Histogram“ pro RGB obraz

### 5.4.2 Řez X-Y

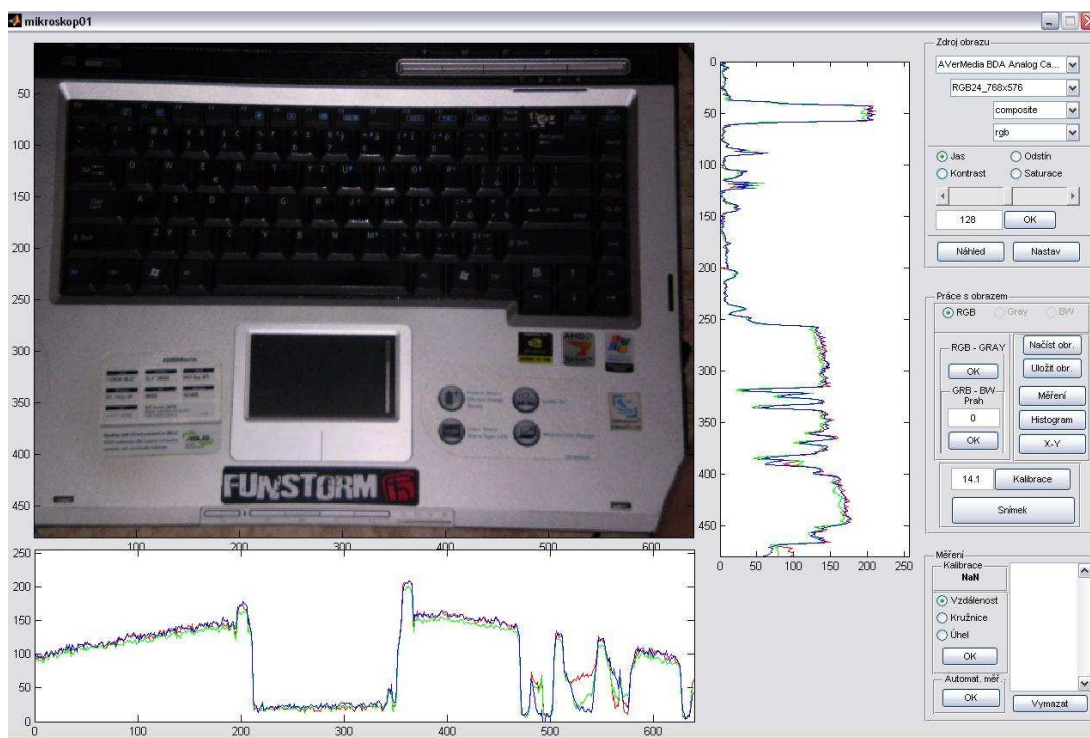
Řez X-Y je funkce, která zjistí a následně zobrazí sled hodnot pixelů v určeném řádku a sloupci. Výpočet je pro každou složku barvy samostatný.

Funkce je umístěna v souboru „fce\_XY.m“.



Obr. 5.9. Vývojový diagram funkce řezu X-Y





Obr. 5.10. Uživatelské prostředí funkce Posloupnosti

### 5.4.3 Uživatelské funkce

Program je pro větší flexibilitu vybaven uživatelskými funkcemi, které může uživatel upravovat aniž by zasahoval do samotného jádra programu.

V každé funkci může uživatel nejen přistupovat ke všem komponentám, měnit jejich parametry a nastavení, ale i vytvářet komponenty nové.

Funkce jsou vytvořeny pomocí m-file souborů s názvy:

- “user1.m”
- “user2.m”
- “user3.m”
- “user4.m”

## 5.5 MĚŘENÍ

Měření vzdáleností je hlavní funkce programu a rozděluje se na tři hlavní části:

- Kalibrace
- Měření s uživatelem
- Automatické měření

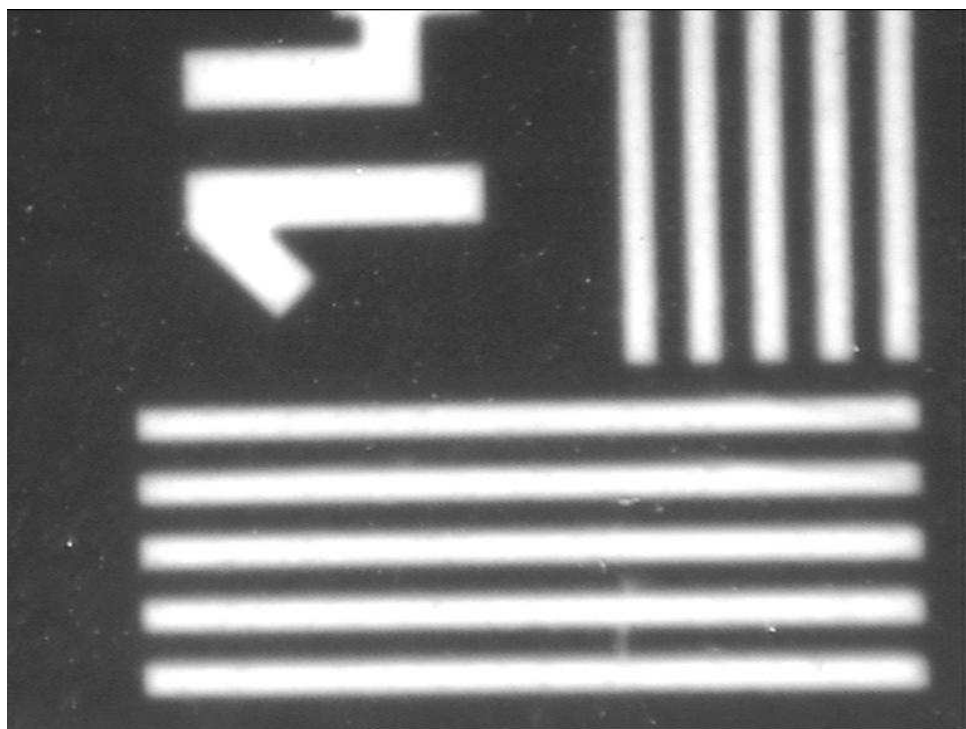
Pro všechna měření je nutné nejdříve provést kalibraci. Po získání kalibračních hodnot pro osy X a Y jsou všechny vzdálenosti převedeny z hodnot pixel do milimetrů.

### 5.5.1 Kalibrace

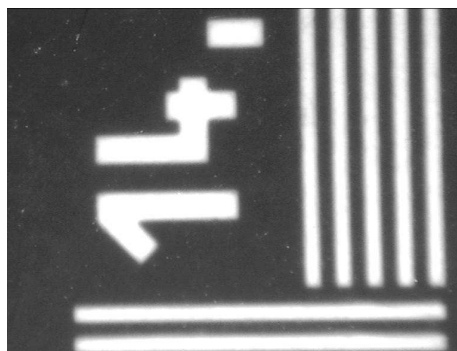
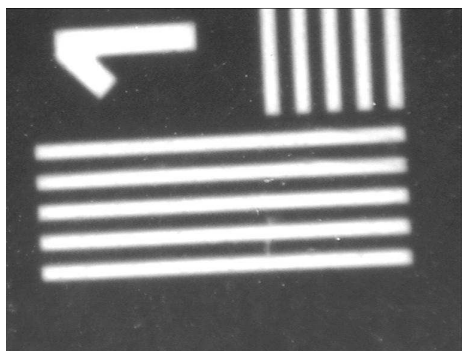
Kalibrace programu spočívá v získání dvou kalibračních konstant  $k_X$  a  $k_Y$ . Vynásobením těchto konstant s velikostí vektoru je získán údaj v metrických jednotkách (mm).

Pro získání kalibračních konstant je nutné nalézt pomocí Houghovy transformace v obou osách dostatečný počet čar na kalibračním vzoru. Pro menší nárok na výpočetní výkon je Houghův prostor nulován pro všechny úhly mimo  $0 \pm 10^\circ$  a  $90 \pm 10^\circ$ . Z velikostí mezer mezi nalezenými čarami je mediánem vypočítána šířka jedné kalibrační čáry na vzoru. Přepočtem ze známé tloušťky v milimetrech je program schopen určit kalibrační konstanty.

Kalibrační vzor je nutné umístit tak, aby všechny vodorovné čáry byly vidět celé a ze svislých čar byla vidět alespoň polovina. Číselný údaj na vzoru být čitelný nemusí.



*Obr. 5.11. Vhodný snímek kalibračního vzoru.*

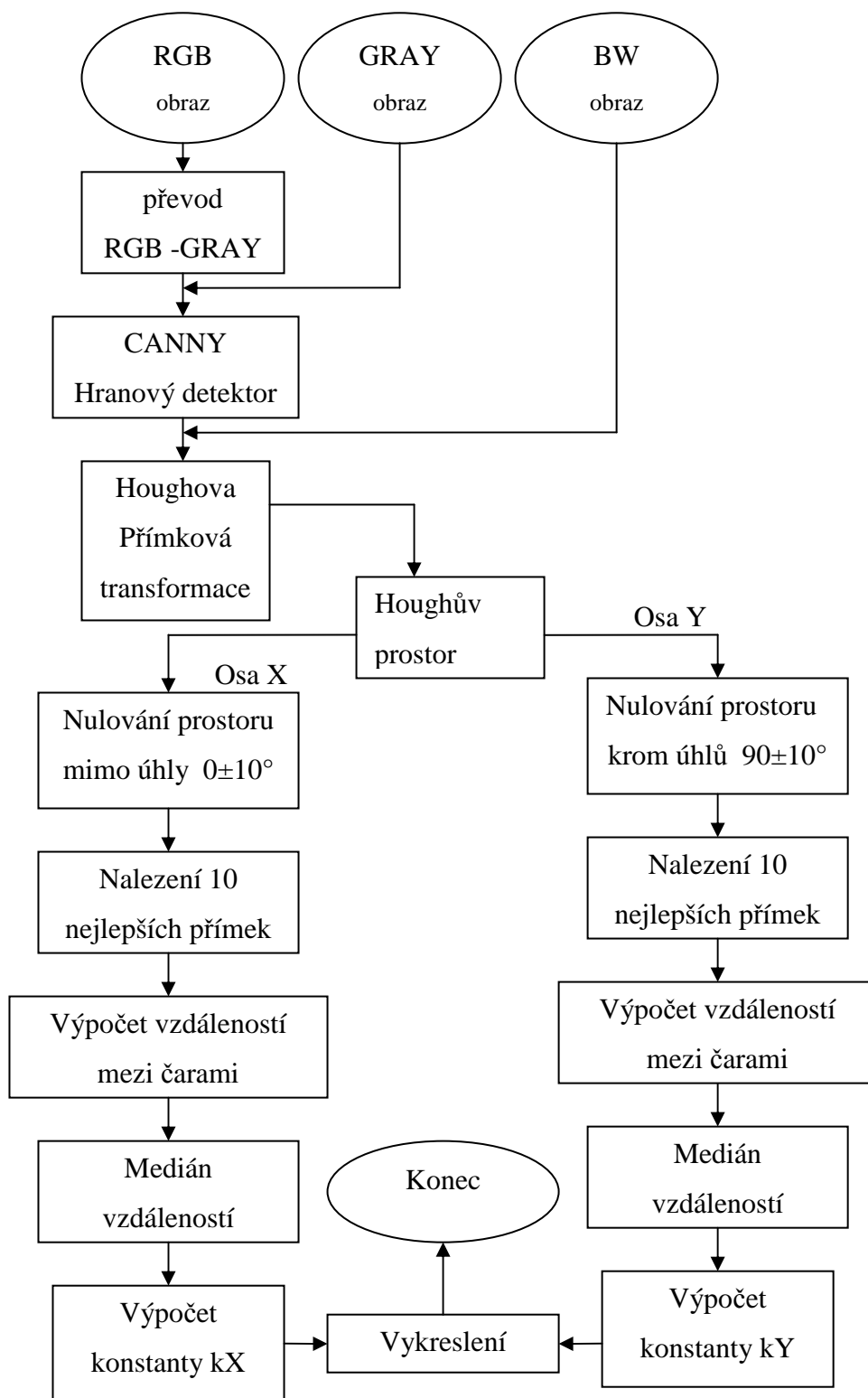


*Obr. 5.12. Nevhodné snímky kalibračního vzoru.*

Výpočet velikosti kalibrační konstanty:

$$kX = \frac{2.4}{d * vzor} [mmpixel^{-1}]$$

kde  $d$  je změřená šířka čáry a  $vzor$  je číslice na kalibračním vzoru.



Obr. 5.13. Vývojový diagram funkce kalibrace

### 5.5.2 Měření s uživatelem

Měření probíhá interaktivně s uživatelem. Uživatel myši určuje měřené body a program vypočítává potřebné hodnoty.

Tyto funkce umožňují uživateli měřit tři základní objekty:

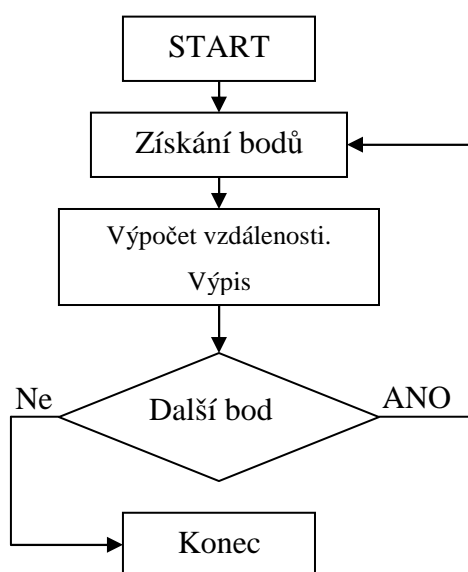
- Vzdálenost
- Kružnice (poloměr, obvod, obsah)
- Úhel

#### Měření vzdáleností:

Měření vzdálenosti spočívá ve výpočtu velikosti vektoru, který určí uživatel označením dvou bodů v měřené rovině.

Velikost vektoru je udaná vztahem:  $|\vec{b}| = \sqrt{(b_1 * kX)^2 + (b_2 * kY)^2} [pixel]$ .

Kde  $b_{12}$  jsou uživatelem určené body a  $kX, kY$  jsou kalibrační konstanty



Obr. 5.14. Vývojový diagram funkce „Měření vzdáleností“

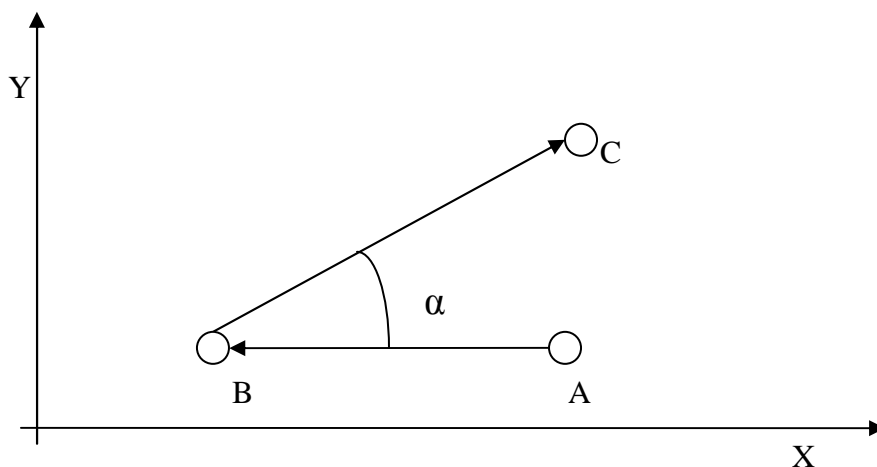


Obr. 5.15. Uživatelské prostředí pro "Měření vzdáleností"

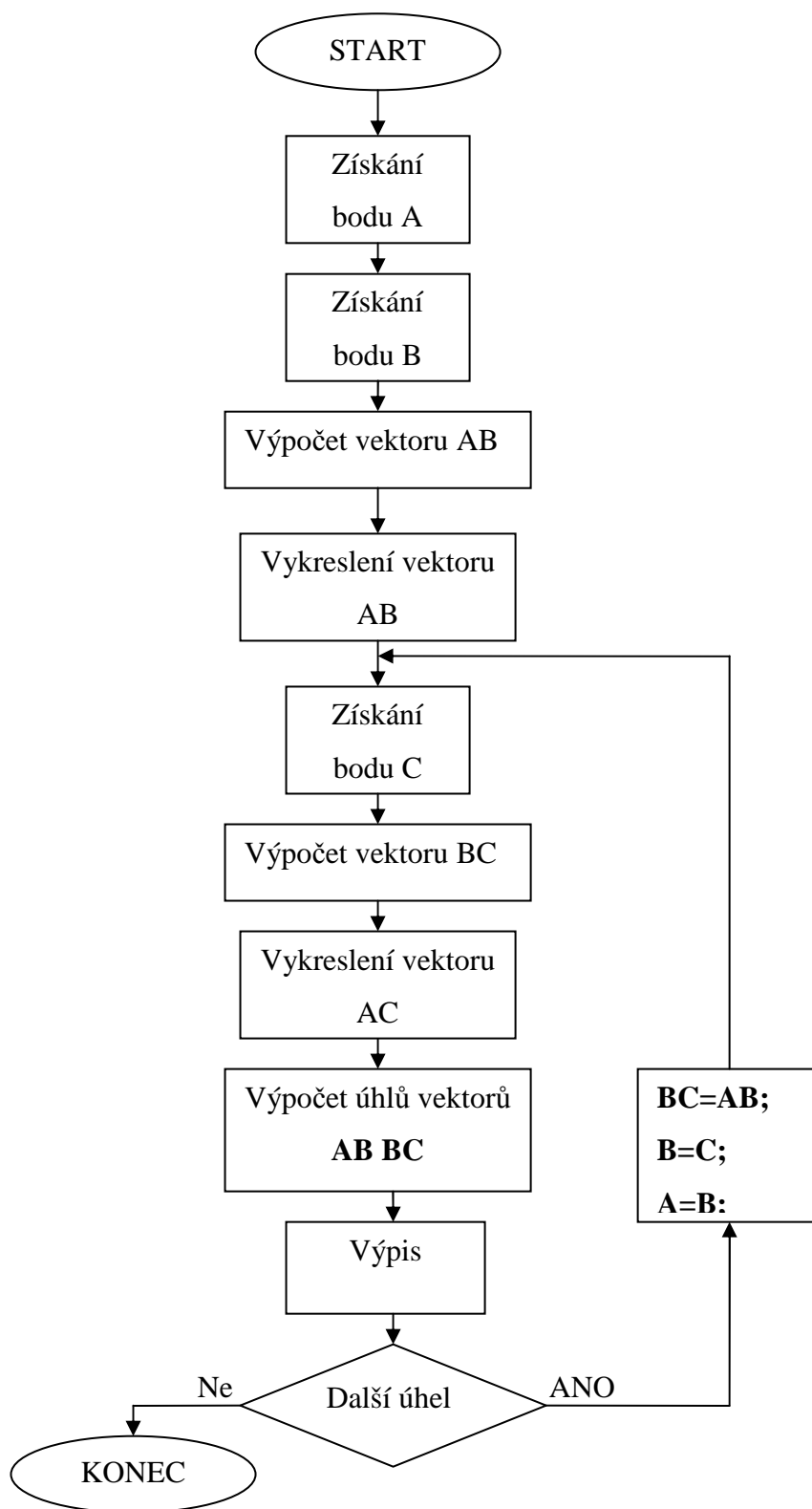
### Měření úhlů:

Uživatel myší určuje trojici bodů A, B, C ze kterých program vypočítá vektory **AB** a **BC**. Z této dvojice vektorů se vztahem  $\cos \alpha = \frac{AB \cdot BC}{|AB| \cdot |BC|}$

Program vypisuje velikost vektoru **AB**, **BC** a úhel  $\alpha$



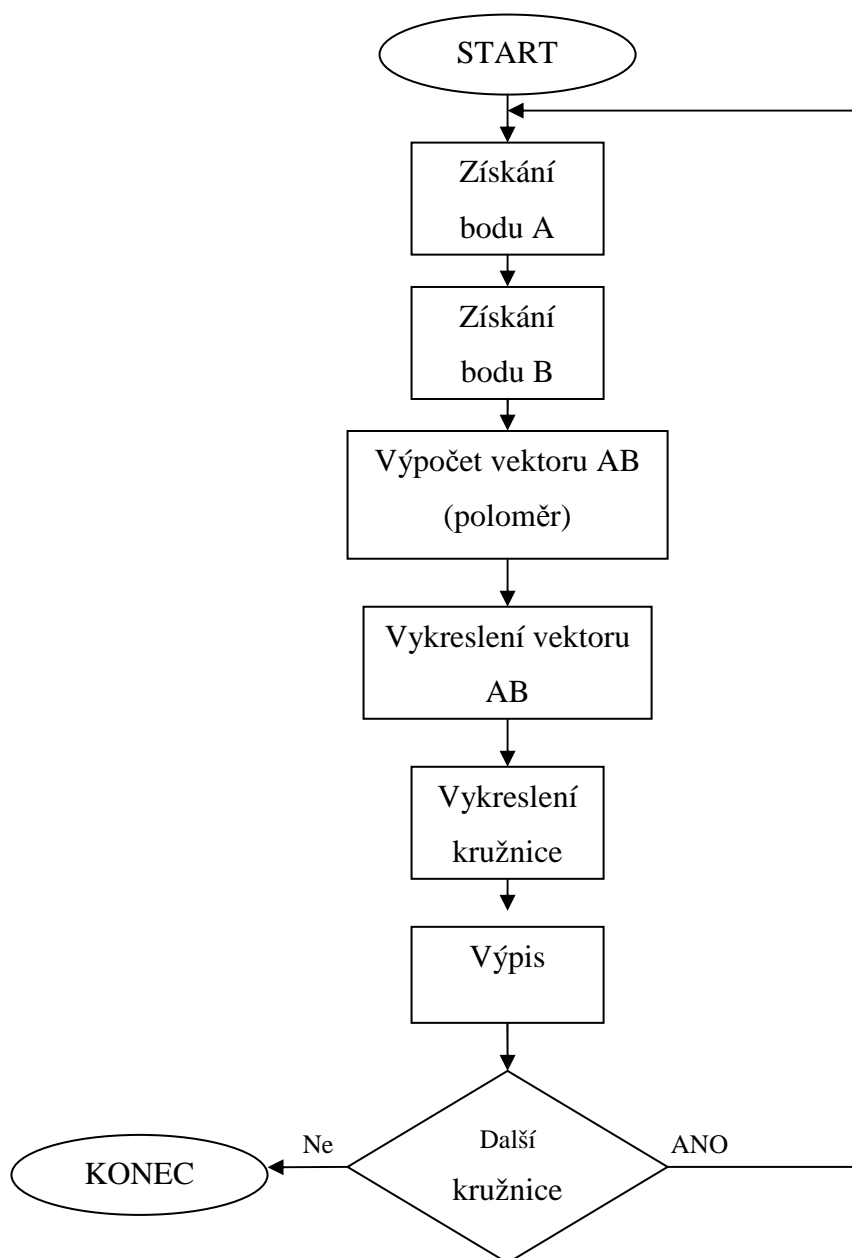
Obr. 5.16. Měření úhlů



Obr. 5.17. Vývojový diagram měření úhlů

### Měření kružnic:

Uživatel myší určuje dvojici bodů A, B, ze kterých program vypočítá velikost vektoru **AB**, který reprezentuje průměr měřené kružnice. Ze známého průměru respektive poloměru program vypočítá ze základních geometrických vzorců obvod kružnice a obsah kruhu jí obepsanou.



Obr. 5.18. Vývojový diagram měření kružnic



### 5.5.3 Automatické měření

Automatické měření v této práci je založeno na Houghově transformaci pro hledání přímk a kružnic. Tyto operace jsou velice náročné na výpočetní výkon PC, proto je nutné věnovat velkou pozornost zpracování obrazu.

Automatické měření se skládá ze dvou skupin:

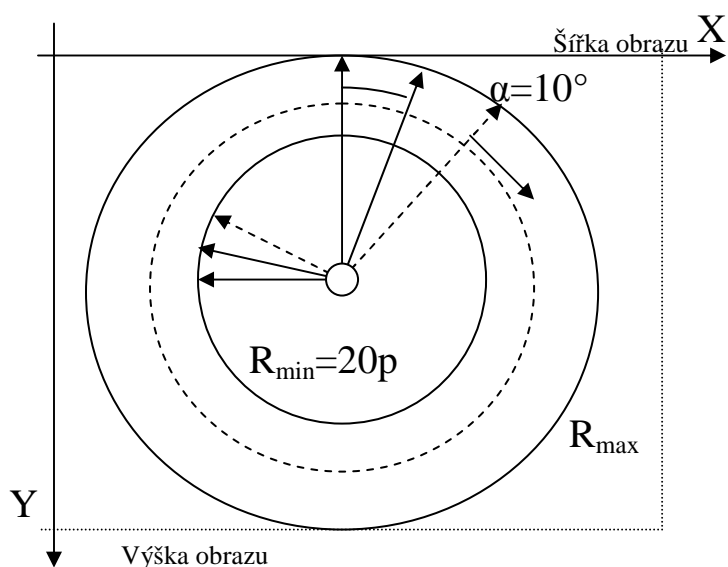
- Hledání kružnic
- Hledání obecných čtyřúhelníků

#### Hledání kružnic:

Po použití Cannyho hranového detektoru jsou z binárního obrazu odstraněny nesouvislé shluky pixelů. Každý nenulový pixel je považován za potencionální bod kružnic. Pro tyto kružnice jsou hledány možné středy a poloměry.

Poloměry se pohybují od intervalu 20 pixelů do poloviny výšky obrazu. Pro každý poloměr nenulového pixelu je hledáno 36 možných středů po  $10^\circ$ .

Pro každý možný střed je inkrementován akumulátorový prostor se souřadnicemi  $X, Y, r$ . Kde  $X, Y$  jsou souřadnice středu hledané kružnice, kde  $r$  je její poloměr.

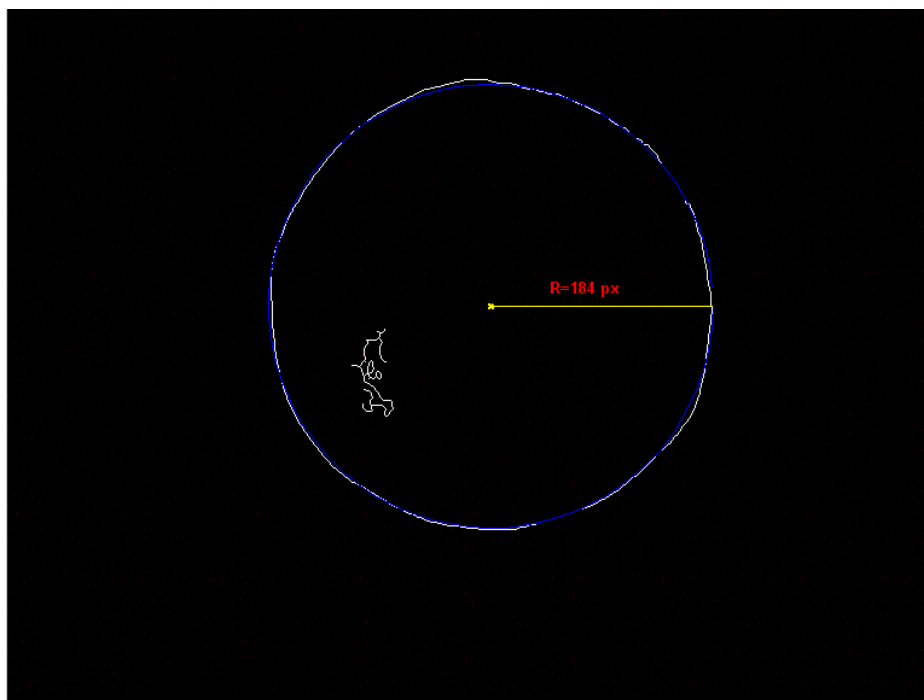


Obr. 5.19. Princip hledání kružnic

*Obr. 5.20. Vývojový diagram hledání kružnic*



*Obr. 5.21. Původní obrázek pro detekci kružnic*

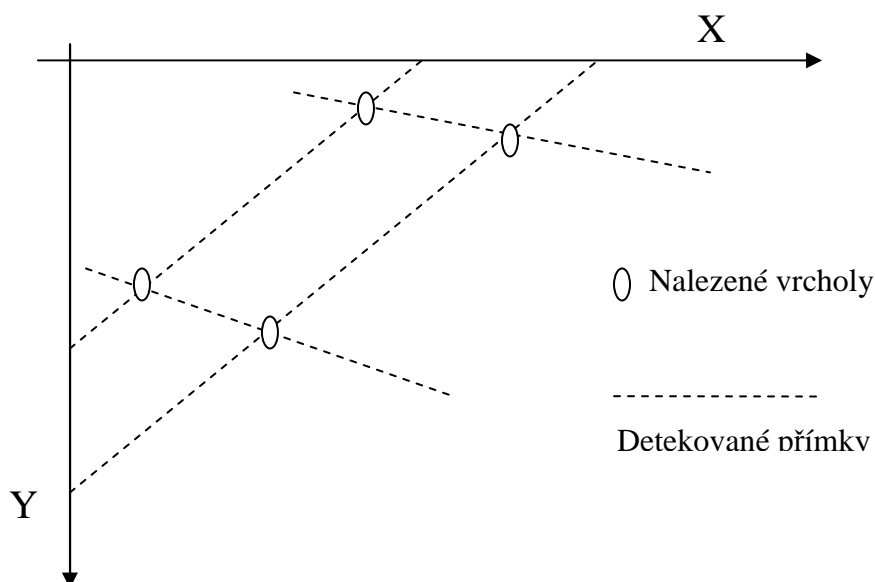


*Obr. 5.22. Nalezený střed a poloměr kružnice*

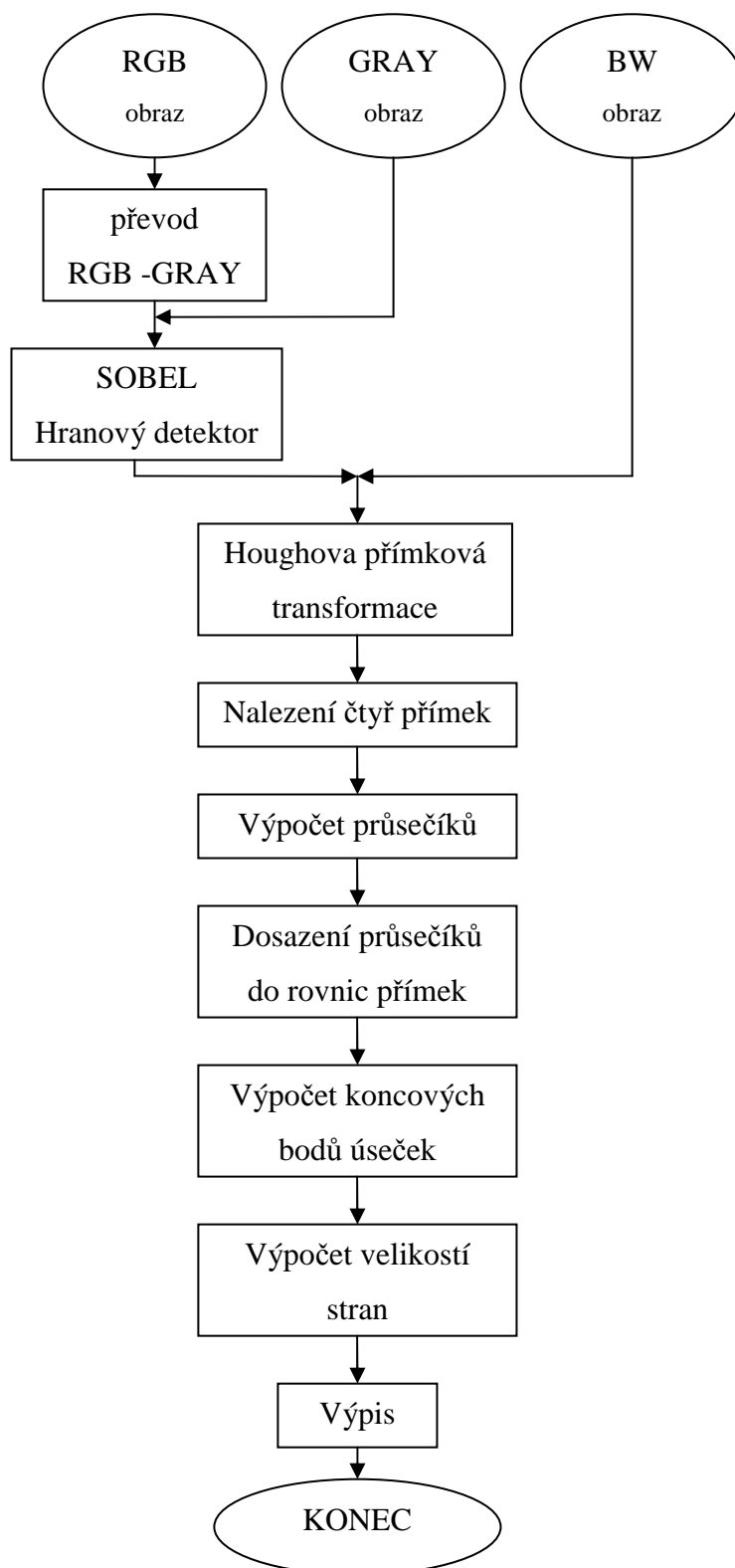
### Hledání obecných $n$ -úhelníků:

Program umožňuje identifikovat a měřit obecné trojúhelníky a čtyřúhelníky. Po použití Sobelova hranového detektoru jsou z binárního obrazu odstraněny nesouvislé shluky pixelů. Po provedení Houghovy transformace jsou v Houghově prostoru nalezeny nejvýznamnější přímky. Průsečíky těchto nalezených přímek jsou považovány za vrcholy identifikovaného polygonu.

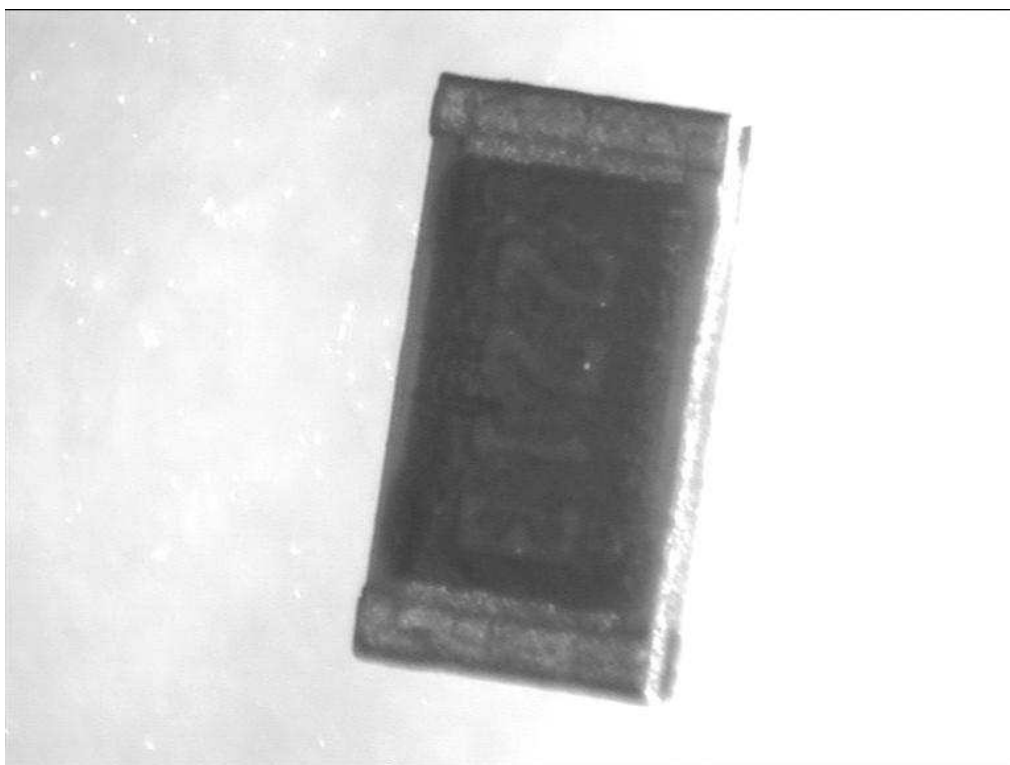
Ze známých vrcholů jsou vypočteny vektory a z nich posléze délky stran.



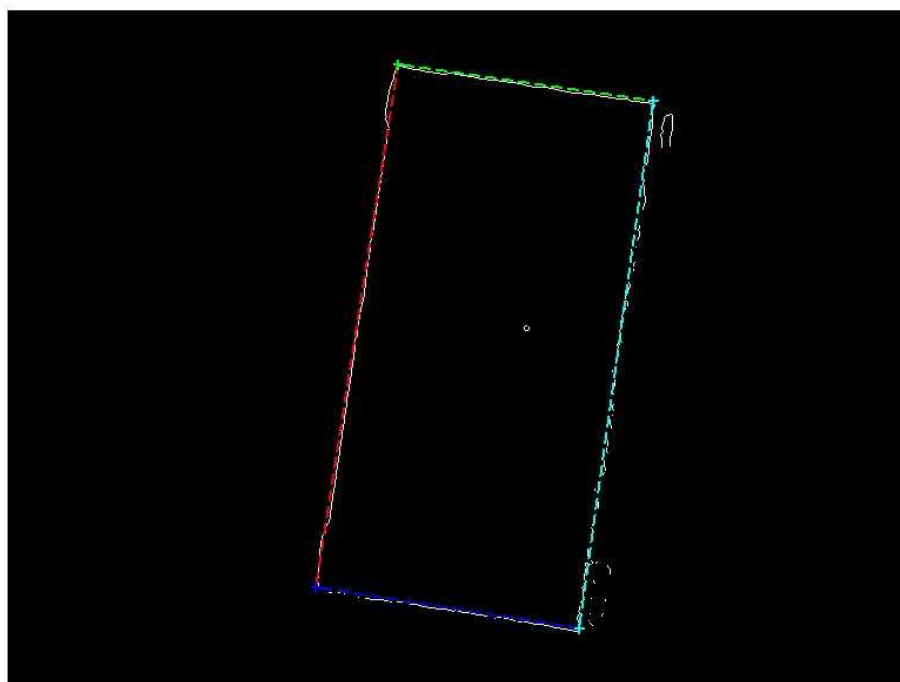
Obr. 5.23. Princip hledání  $n$ -úhelníků



Obr. 5.24. Vývojový diagram hledání n-úhelníků



*Obr. 5.25. Původní obraz pro detekci čtyřúhelníku*



*Obr. 5.26. Nalezený čtyřúhelník*

## **6. ZÁVĚR**

Uživatel tohoto programu je schopen manipulovat s obrazem i videem. Lze nakonfigurovat libovolný hardware k dosažení optimálního obrazu. Automaticky identifikovat jednoduché tvary a složitější s operátorem.

Díky kalibrační funkci, která je v programu obsažena, je možné převést naměřené údaje z pixelového do metrického rozměru.

Pomocí implementovaných funkcí lze provádět základní úkony s obrazem. Pro větší komfort může uživatel využít tzv. „Uživatelských funkcí“, poskytujících široký prostor pro jakékoliv uživatelské požadavky.

## 7. POUŽITÁ LITERATURA

- [1] The Mathworks- MATLAB and Simulink for technical Computing [online]  
2007, [cit 21.12.2007] <http://www.mathworks.com>
- [2] Image Transforms- Hough Transform [online]  
2007, [cit 21.12.2007] <http://homepages.inf.ed.ac.uk/>
- [3] Karel Horák, Kalová Ilona: *Počítačové vidění*, Brno, 1.11.2006
- [4] Hlaváč V., Šonka M.: *Počítačové vidění*, Grada 1992, ISBN 80-85424-67-3



## **8. PŘÍLOHY**

### **8.1 OBSAH CD**

- kompletní program
- bakalářská práce v elektronické podobě
- obrázky se změřenými objekty

## 8.2 IDENTIFIKOVANÉ OBJEKTY

